Bảng tóm tắt khoa học dữ liệu được biên soạn bởi Maverick Lin (http://mavericklin.com)

Cập nhật lần cuối vào ngày 13 tháng 8 năm 2018

Khoa học dữ liệu là gì?

Lĩnh vực đa ngành tập hợp các khái niệm từ khoa học máy tính, thống kê/học máy và phân tích dữ liệu để hiểu và rút ra những hiểu biết sâu sắc từ lượng dữ liệu ngày càng tăng.

Hai mô hình nghiên cứu dữ liệu.

1. Dựa trên giả thuyết: Đưa ra một vấn đề, loại vấn đề gì dữ liệu nào chúng ta cần để giúp giải quyết nó?
2. Dựa trên dữ liệu: Với một số dữ liệu, những vấn đề thú vị nào có thể được giải quyết với nó?

Trọng tâm của khoa học dữ liệu là luôn đặt câu hỏi. Luôn tò mò về thế giới.

1. Chúng ta có thể học được gì từ dữ liệu này?
2. Chúng ta có thể thực hiện những hành động nào khi tìm thấy thứ chúng ta đang tìm kiếm?

Các loại dữ liệu

Có cấu trúc: Dữ liệu có cấu trúc được xác định trước. ví dụ: bảng, bảng tính hoặc cơ sở dữ liệu quan hệ.
Dữ liệu phi cấu trúc: Dữ liệu không có cấu trúc được xác định trước, có kích thước hoặc dạng bất kỳ, không thể dễ dàng lưu trữ trong bảng. ví dụ: các đốm văn bản, hình ảnh,
âm thanh Dữ liệu định lượng: Số. ví dụ: chiều cao, cân nặng
Dữ liệu phân loại: Dữ liệu có thể được gắn nhãn hoặc chia thành các nhóm. ví dụ như chủng tộc, giới tính, màu tóc.
Dữ liệu lớn: Bộ dữ liệu khổng lồ hoặc dữ liệu chứa sự đa dạng hơn với khối lượng ngày càng tăng và tốc độ ngày càng cao (3 Vs). Không thể vừa với bộ nhớ của một máy.

Nguồn dữ liệu/Fomat

Các định dạng dữ liệu phổ biến nhất CSV, XML, SQL, JSON, Bộ đệm giao thức
Nguồn dữ liệu Công ty/Dữ liệu độc quyền, API, Chính phủ, Học thuật, Quét/Thu thập dữ liệu web

Các loại vấn đề chính

Hai vấn đề phát sinh nhiều lần trong khoa học dữ liệu.
Phân loại: Gán một cái gì đó vào một tập hợp các khả năng riêng biệt. ví dụ như thư rác hoặc không phải thư rác, Đảng Dân chủ hoặc Cộng hòa, nhóm máu (A, B, AB, O)
Hồi quy: Dự đoán một giá trị số. ví dụ: thu nhập của ai đó, GDP năm tới, giá cổ phiếu

Tổng quan về xác suất

Lý thuyết xác suất cung cấp một khuôn khổ lý luận về khả năng xảy ra của các sự kiện.

Thuật ngữ Thí

nghiệm: quy trình mang lại một trong các tập hợp kết quả có thể xảy ra, ví dụ như tung liên tục một con súc sắc hoặc đồng xu Không gian mẫu S: tập hợp các kết quả có thể xảy ra của một thí nghiệm, ví dụ: nếu tung một con xúc xắc, S = (1,2,3,4, 5,6 Sự kiện E: tập hợp các kết quả của một thử nghiệm, ví dụ sự kiện cuộn là 5 hoặc sự kiện tổng của 2 cuộn là 7 Xác suất của Kết quả s hoặc P(s): số thỏa mãn 2
tính chất 1. cho mỗi tính chất kết quả
P(s) 1 sp,(s∅) ≤ 1 2.

Xác suất của Sự kiện E: tổng xác suất của các kết quả của thí nghiệm: p(E) = s E p(s)
Biến ngẫu nhiên V: hàm số ở đầu ra
xuất phát từ một không gian xác suất
Giá trị kỳ vọng của biến ngẫu nhiên V: E(V) = p(s) * V(s)
s S

Độc lập, có điều kiện, hợp chất

Biến cố độc lập: A và B độc lập nếu:

$$P(A \cap B) = P(A)P(B)$$
$$P(A|B) = P(A)$$
$$P(B|A) = P(B)$$

Xác suất có điều kiện: P(A|B) = P(A,B)/P(B)
Định lý Bayes: P(A|B) = P(B|A)P(A)/P(B)
Xác suất chung: P(A,B) = P(B|A)P(A)
Xác suất cận biên: P(A)

Phân bố xác suất Hàm mật độ xác

suất (PDF) Cung cấp xác suất mà một rv nhận giá trị x: pX(x) = P(X = x)
Hàm mật độ tích lũy (CDF) Cho biết xác suất một biến ngẫu nhiên nhỏ hơn hoặc bằng x: FX(x) = P(X ≤ x)

Lưu ý: Tệp PDF và CDF của một biến ngẫu nhiên nhất định chứa thông tin giống hệt nhau.

Thống kê mô tả

Cung cấp cách thu thập một tập dữ liệu hoặc mẫu nhất định.
Có hai loại chính: tính trung tâm và tính biến đổi đo.

Trung bình

số học trung tâm Hữu ích để mô tả các phân bố đối xứng không có giá trị ngoại lệ μX = Trung bình $\frac{1}{N}$ x
hình học Hữu ích cho các tỷ lệ trung bình. Luôn nhỏ hơn trung bình số học = √n a1a2...a3 Trung vị Giá trị trung bình chính xác giữa một tập dữ liệu. Hữu ích cho việc phân phối sai lệch hoặc dữ liệu có các giá trị ngoại lệ.
Chế độ Phần tử thường xuyên nhất trong tập dữ liệu.

Sự biến đổi

Độ lệch chuẩn Đo chênh lệch bình phương giữa các phần tử riêng lẻ và giá trị trung bình i=1(xi x)

$$\sigma = \frac{\overline{N \quad - 2}}{N \ 1}$$

Phương sai V = σ 2

Giải thích phương sai

Phương sai là một phần cố hữu của vũ trụ. Không thể thu được kết quả tương tự sau khi quan sát lặp đi lặp lại cùng một sự kiện do nhiễu/lỗi ngẫu nhiên. Phương sai có thể được giải thích bằng cách quy cho lỗi lấy mẫu hoặc đo lường. Những lần khác, phương sai là do những biến động ngẫu nhiên của vũ trụ.

Phân tích tương quan Hệ

số tương quan r(X,Y) là một thống kê đo lường mức độ Y là hàm của X và ngược lại.
Các giá trị tương quan nằm trong khoảng từ -1 đến 1, trong đó 1 nghĩa là tương quan hoàn toàn, -1 nghĩa là tương quan nghịch và 0 nghĩa là không tương quan.
Hệ số Pearson Đo lường mức độ mối quan hệ giữa các biến liên quan tuyến tính

$$r = \frac{Cov(X,Y)}{\sigma(X)\sigma(Y)}$$

Hệ số xếp hạng Spearman Được tính theo cấp bậc và mô tả các mối quan hệ đơn điệu

Lưu ý: Tương quan không hàm ý quan hệ nhân quả!

Làm sạch dữ liệu

Làm sạch dữ liệu là quá trình biến dữ liệu thô thành tập dữ liệu sạch và có thể phân tích được. “Rác vào, rác ra.” Hãy chắc chắn rằng rác không được đưa vào.

Lỗi so với hiện vật

1. Lỗi: thông tin bị mất trong quá trình thu thập và không bao giờ có thể khôi phục lại được, ví dụ như mất điện, máy chủ bị hỏng
2. Hiện vật: các

sự cố hệ thống phát sinh từ quá trình làm sạch dữ liệu. những vấn đề này có thể được sửa chữa nhưng trước tiên chúng ta phải khám phá chúng

Khả năng tương thích dữ liệu

Các vấn đề tương thích dữ liệu phát sinh khi hợp nhất các tập dữ liệu. Hãy chắc chắn rằng bạn đang so sánh "táo với táo" chứ không phải "táo với cam". Các loại chuyển đổi/hợp nhất chính: • đơn vị (số liệu so với hệ đo lường Anh) • số

(số thập phân so với số nguyên), •
tên (John Smith so với Smith, John), • thời
gian/ngày (UNIX so với UTC so với GMT), • tiền tệ
(loại tiền tệ, điều chỉnh theo lạm phát, chia

vết lõm)

Xử lý dữ liệu

Quá trình xử lý các giá trị bị thiếu. Các phương pháp thích hợp phụ thuộc vào loại dữ liệu chúng tôi đang làm việc. Các phương pháp chung bao gồm:

• Loại bỏ tất cả các bản ghi chứa dữ liệu bị thiếu • Dựa trên suy nghiệm: đưa ra dự đoán hợp lý dựa trên
kiến thức về miền cơ bản • Giá trị trung
bình: điền giá trị trung bình vào dữ liệu còn thiếu • Giá trị ngẫu nhiên

• Hàng xóm gần nhất: điền dữ liệu bị thiếu bằng cách sử dụng các điểm dữ liệu
tương tự • Nội suy: sử dụng phương pháp như hồi quy tuyến tính để
dự đoán giá trị của dữ liệu bị thiếu

Phát hiện ngoại lệ

Các ngoại lệ có thể cản trở việc phân tích và thường phát sinh từ những sai sót trong quá trình thu thập dữ liệu. Việc thực hiện "kiểm tra độ tỉnh táo" là điều hợp lý.

Điều khoản khác

Viết thường, loại bỏ không phải chữ và số, sửa chữa, unidecode, loại bỏ các ký tự không xác định

Lưu ý: Khi làm sạch dữ liệu, hãy luôn duy trì cả dữ liệu thô và (các) phiên bản đã được làm sạch. Dữ liệu thô phải được giữ nguyên và bảo quản để sử dụng trong tương lai. Bất kỳ loại làm sạch/phân tích dữ liệu nào cũng phải được thực hiện trên bản sao của dữ liệu thô.

Kỹ thuật tính năng

Kỹ thuật tính năng là quá trình sử dụng kiến thức miền để tạo ra các tính năng hoặc biến đầu vào giúp thuật toán học máy hoạt động tốt hơn. Thực hiện chính xác, nó có thể giúp tăng khả năng dự đoán cho các mô hình của bạn.
Kỹ thuật tính năng là một nghệ thuật hơn là khoa học. FE là một trong những bước quan trọng nhất để tạo ra một mô hình tốt.
Như Andrew Ng đã nói:

> “Việc tạo ra các tính năng rất khó, tốn thời gian, đòi hỏi phải có kiến thức chuyên môn. 'Học máy ứng dụng' về cơ bản là kỹ thuật tính năng.”

Dữ liệu liên tục

Các thước đo thô: dữ liệu chưa được chuyển đổi Làm tròn: đôi khi độ chính xác là nhiễu; làm tròn đến số nguyên gần nhất, số thập phân, v.v.

Chia tỷ lệ: log, điểm z, thang đo tối thiểu Phép
tính: điền vào các giá trị còn thiếu bằng cách sử dụng giá trị trung bình, trung vị, đầu ra mô hình, v.v.
Binning: chuyển đổi các đối tượng số thành các đối tượng phân loại (hoặc được đánh dấu), ví dụ: các giá trị trong khoảng 1-10 thuộc về A, trong khoảng 10-20 thuộc về B, v.v.
Tương tác: tương tác giữa các tính năng: ví dụ: phép kéo phụ, phép cộng, phép nhân, kiểm tra thống kê Thống kê: biến đổi log/
power (giúp biến các phân phối lệch trở nên bình thường hơn), Thống kê hàng Box-Cox: số lượng NaN, 0, giá trị âm,
tối đa, tối thiểu , v.v. Giảm kích thước: sử dụng PCA, phân cụm, phân tích nhân tố, v.v.

Dữ liệu rời rạc

Mã hóa: do một số thuật toán ML không thể hoạt động trên dữ liệu phân loại, chúng ta cần biến dữ liệu phân loại thành dữ liệu số hoặc vectơ

Giá trị thứ tự: chuyển đổi từng tính năng riêng biệt thành một chuỗi

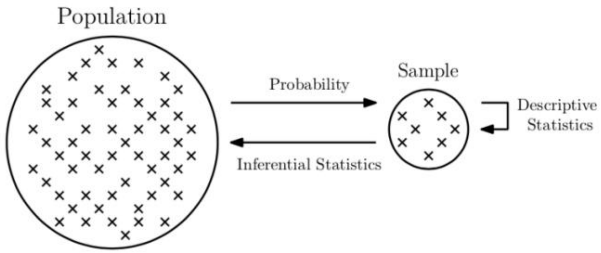
số dom (ví dụ [r,g,b] trở thành [1,2,3])
Mã hóa một lần nóng: mỗi tính năng m trở thành một vectơ có độ dài m chỉ chứa một 1 (ví dụ: [r, g, b] trở thành [[1,0,0],[0,1,0],[0 ,0,1]])

Lược đồ băm tính năng: biến các tính năng tùy ý thành các chỉ mục trong vectơ hoặc ma trận. Nhúng: nếu sử

dụng từ, hãy chuyển đổi từ thành vectơ (nhúng từ)

Phân tích thống kê

Quá trình suy luận thống kê: có một tập hợp cơ bản những thứ có thể xảy ra mà chúng ta có thể quan sát được và chỉ một tập hợp con nhỏ trong số đó được lấy mẫu thực sự (lý tưởng là ngẫu nhiên). Lý thuyết xác suất mô tả những đặc tính mà mẫu của chúng ta lẽ ra phải có các đặc tính của tổng thể, nhưng suy luận thống kê cho phép chúng ta suy ra toàn bộ tổng thể sẽ như thế nào sau khi phân tích mẫu.



Lấy mẫu từ các phân phối

Lấy mẫu biến đổi nghịch đảo Các điểm lấy mẫu từ một phân bố xác suất nhất định đôi khi cần thiết để chạy mô phỏng hoặc liệu dữ liệu của bạn có phù hợp với một phân phối cụ thể hay không. Kỹ thuật chung được gọi là lấy mẫu biến đổi nghịch đảo hoặc biến đổi Smirnov. Đầu tiên vẽ một số ngẫu nhiên p giữa [0,1]. Tính giá trị x sao cho CDF bằng p: FX(x) = p. Sử dụng x làm giá trị là giá trị ngẫu nhiên được rút ra từ phân phối

được mô tả bởi FX(x).

Lấy mẫu Monte Carlo Ở các chiều cao hơn, việc lấy mẫu chính xác từ một phân phối nhất định trở nên phức tạp hơn.
Nói chung muốn sử dụng các phương pháp Monte Carlo, thường tuân theo các quy tắc sau: xác định miền của các đầu vào có thể, tạo đầu vào ngẫu nhiên từ phân bố xác suất trên miền, thực hiện phép tính xác định và phân tích kết quả.

Phân phối thống kê cổ điển

Phân phối nhị thức (rời rạc)

Giả sử X được phân phối Bin(n,p). X là số lần "thành công" mà chúng ta sẽ đạt được trong n lần thử độc lập, trong đó mỗi lần thử là thành công hoặc thất bại và mỗi lần thành công đều xảy ra với cùng xác suất p và mỗi lần thất bại xảy ra với xác suất q=1-p.

PDF: $P(X = x) = x\ (1 \quad {}^{N}_{k}p)\ n\ \ x\ p$ EV: μ = np Phương sai = npq

Phân phối chuẩn/Gaussian (Liên tục)

Giả sử X trong phân bố N (μ, σ2). Đó là một sự phân bố hình chuông và đối xứng. Phần lớn các giá trị nằm gần giá trị trung bình và không có giá trị nào quá cực đoan. Tổng quát hóa phân bố nhị thức là n ∞. (x μ) 2/2σ 2 PDF: P(x) =

$$\frac{1}{e\ \sigma\ \sqrt{\ }2p}$$

EV: μ Phương sai: σ Ý ^{2}

nghĩa: quy tắc 68%-95%-99%. 68% khối lượng xác suất nằm trong khoảng 1σ giá trị trung bình, 95% nằm trong khoảng 2σ và 99,7% nằm trong khoảng 3σ.

Phân phối Poisson (rời rạc)

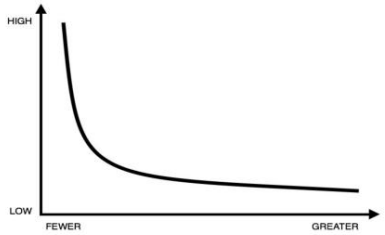
Giả sử X được phân phối Pois(λ). Poisson biểu thị xác suất của một số sự kiện nhất định xảy ra trong một khoảng thời gian/không gian cố định nếu các sự kiện này xảy ra độc lập và với tốc độ không đổi đã biết λ.

PDF: $P(x) = \frac{e\ ll\ x}{x!}$ EV: λ Phương sai = λ

Phân phối luật quyền lực (rời rạc)

Nhiều phân phối dữ liệu có đuôi dài hơn nhiều so với phân phối bình thường hoặc phân phối Poisson. Nói cách khác, sự thay đổi của một đại lượng thay đổi theo lũy thừa của một đại lượng khác. Nó giúp đo lường sự bất bình đẳng trên thế giới. ví dụ: sự giàu có, tần suất từ và Nguyên tắc Pareto (Quy tắc 80/20)

PDF: P(X=x) = cx trong đó Một α,là số mũ của định luật và c là hằng số chuẩn hóa



Mô hình hóa- Tổng quan

Mô hình hóa là quá trình kết hợp thông tin vào một công cụ có thể dự báo và đưa ra dự đoán. Thông thường, chúng ta đang xử lý mô hình thống kê trong đó chúng ta muốn phân tích mối quan hệ giữa các biến. Về mặt hình thức, chúng ta muốn ước lượng hàm f(X) sao cho:

$$Y = f(X) +$$

trong đó X = (X1, X2, ...Xp) đại diện cho biến đầu vào, Y đại diện cho biến đầu ra và đại diện cho biến ngẫu nhiên lỗi.

Học thống kê là tập hợp các phương pháp để ước tính f(X) này.

Tại sao ước tính f(X)?

Dự đoán: khi chúng ta có ước tính tốt ^f(X), chúng ta có thể sử dụng nó để đưa ra dự đoán về dữ liệu mới. Chúng tôi coi ^f như một hộp đen, vì chúng tôi chỉ quan tâm đến tính chính xác của các dự đoán chứ không quan tâm đến lý do hoặc cách thức hoạt động của nó.

Suy luận: chúng ta muốn hiểu mối quan hệ giữa X và Y. Chúng ta không thể coi ^f là hộp đen nữa vì chúng ta muốn hiểu Y thay đổi như thế nào đối với X = (X1, X2, ...Xp)

Thông tin thêm về

Phần sai số bao gồm sai số có thể giảm được và không thể giảm được, điều này sẽ khiến chúng ta không bao giờ có được ước tính ^f hoàn hảo. • Có

thể giảm được: lỗi có thể được giảm bớt bằng cách sử dụng kỹ thuật học thống kê thích hợp nhất để ước tính f. Mục đích là giảm thiểu sai số có thể giảm được.

• Không thể giảm được: lỗi không thể giảm được không

vấn đề là chúng ta ước tính f tốt đến mức nào. Lỗi không thể giảm được là không xác định và không thể đo lường được và sẽ luôn là giới hạn trên của .

Lưu ý: Sẽ luôn có sự cân bằng giữa tính linh hoạt của mô hình (dự đoán) và khả năng diễn giải mô hình (suy luận). Đây chỉ là một trường hợp khác của sự đánh đổi sai lệch-phương sai. Thông thường, khi tính linh hoạt tăng lên thì khả năng diễn giải sẽ giảm xuống. Phần lớn việc học/mô hình hóa thống kê đang tìm cách cân bằng cả hai.

Mô hình hóa- Triết học

Mô hình hóa là quá trình kết hợp thông tin vào một công cụ có thể dự báo và đưa ra dự đoán.

Việc thiết kế và xác nhận các mô hình cũng như đánh giá hiệu suất của các mô hình là rất quan trọng. Lưu ý rằng mô hình dự báo tốt nhất có thể không phải là mô hình chính xác nhất.

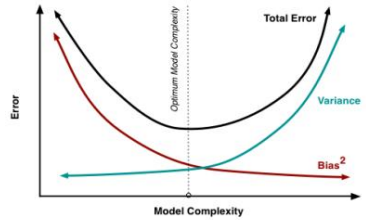
Các triết lý mô hình hóa Occam's

Razor Nguyên tắc triết học cho rằng lời giải thích đơn giản nhất là lời giải thích tốt nhất. Trong mô hình hóa, nếu được cho hai mô hình có khả năng dự đoán tốt như nhau thì chúng ta nên chọn mô hình đơn giản hơn. Việc chọn cái phức tạp hơn thường có thể dẫn đến việc trang bị quá mức.

Sự đánh đổi phương sai thiên vị Một phần vốn có của mô hình dự đoán, trong đó các mô hình có độ lệch thấp hơn sẽ có phương sai cao hơn và ngược lại. Mục tiêu là đạt được độ lệch thấp và phương sai

thấp. • Độ lệch: lỗi từ các giả định không chính xác để làm cho chức năng tar-get dễ học hơn (độ lệch cao thiếu các mối quan hệ liên quan hoặc không

phù hợp) • Phương sai: lỗi do độ nhạy đối với các dao động trong tập dữ liệu hoặc mức độ ước tính mục tiêu sẽ khác nhau nếu khác nhau dữ liệu huấn luyện đã được sử dụng (phương sai cao nhiễu mô hình hoặc trang bị quá mức)



Định lý Không có bữa trưa miễn phí Không có thuật toán học máy nào tốt hơn tất cả các thuật toán khác trong mọi vấn đề.

Người ta thường thử nhiều mô hình và tìm ra mô hình phù hợp nhất cho một vấn đề cụ thể.

Suy nghĩ như Nate Silver

1. Suy nghĩ theo xác suất Các dự báo có ý nghĩa hơn các tuyên bố cụ thể và phải được báo cáo dưới dạng phân bố xác suất (bao gồm σ cùng với dự đoán trung bình μ.

2. Kết hợp thông tin mới Sử dụng các mô hình trực tiếp, liên tục cập nhật bằng thông tin mới. Để cập nhật, hãy sử dụng lý luận Bayes để tính toán xác suất thay đổi như thế nào trước những bằng chứng mới.

3. Tìm kiếm dự báo đồng thuận Sử dụng nhiều nguồn bằng chứng khác nhau. Một số mô hình hoạt động theo cách này, chẳng hạn như tăng cường và đóng bao, sử dụng số lượng lớn các bộ phân loại yếu để tạo ra một mô hình mạnh.

Mô hình hóa- Phân loại

Có nhiều loại mô hình khác nhau. Điều quan trọng là phải hiểu sự cân bằng và khi nào nên sử dụng một giải pháp nhất định loại mô hình.

Tham số so với không tham số • Tham số:
các mô hình đầu tiên đưa ra giả định về dạng hàm hoặc hình dạng của f (tuyến tính). Sau đó phù hợp với mô hình. Điều này làm giảm việc ước lượng f thành chỉ ước lượng tập hợp các tham số, nhưng nếu giả định của chúng ta sai sẽ dẫn đến kết quả không tốt.
• Phi tham số: các mô hình không đưa ra bất kỳ giả định nào về f, điều này cho phép chúng phù hợp với phạm vi hình dạng rộng hơn; nhưng có thể dẫn đến việc khớp quá mức

Giám sát so với Không giám sát • Được
giám sát: các mô hình khớp các biến đầu vào xi = (x1, x2, ...xn) với các biến đầu ra đã biết yi = (y1, y2, ...yn)
• Không giám
sát: các mô hình nhận các biến đầu vào xi = (x1, x2, ...xn) nhưng không có đầu ra liên kết để giám sát quá trình huấn luyện. Mục tiêu là hiểu mối quan hệ giữa các biến hoặc quan sát.

Hộp đen so với mô tả
• Hộp đen: các mô hình đưa ra quyết định, nhưng chúng tôi không biết điều gì xảy ra "dưới mui xe" ví dụ như học sâu, mạng lưới thần kinh • Mô tả:
các mô hình cung cấp cái nhìn sâu sắc về lý do tại sao họ đưa ra quyết định, ví dụ như hồi quy tuyến tính, cây quyết định
Nguyên tắc
đầu tiên so với Dựa trên dữ liệu • Nguyên
tắc đầu tiên: các mô hình dựa trên niềm tin trước đó về cách hệ thống đang được điều tra hoạt động, kết hợp kiến thức miền (ad-hoc) • Dựa trên dữ liệu: các
mô hình dựa trên mối tương quan được quan sát sự khác biệt giữa các biến đầu vào và đầu ra

Xác định so với ngẫu nhiên
• Có tính tất định: các mô hình tạo ra một ”tiền diction” ví dụ có hoặc không, đúng hay sai
• Stochastic: mô hình tạo ra phân bố xác suất quan điểm về các sự kiện có thể
xảy ra Flat vs. Hierarchical
• Flat: mô hình giải quyết vấn đề ở một cấp độ duy nhất, không có khái niệm về các vấn đề
con • Hierarchical: mô hình giải quyết một số vấn đề khác nhau các bài toán con lồng nhau

Mô hình hóa-Đánh giá số liệu

Cần phải xác định mô hình của chúng tôi tốt như thế nào. Cách tốt nhất để đánh giá các mô hình là dự đoán ngoài mẫu (các điểm dữ liệu mà mô hình của bạn chưa từng thấy).

Phân loại

	Dự đoán Có	Dự đoán Không
Thực tế Có	Kết quả tích cực thực sự (TP)	Âm tính giả (FN)
Thực tế Không có	Kết quả dương tính giả (FP)	Kết quả âm tính thực sự (TN)

Độ chính xác: tỷ lệ dự đoán đúng trên tổng số dự đoán. Gây hiểu lầm khi quy mô lớp học rất khác nhau. độ chính xác = Độ chính xác: tần suất bộ phân loại $\frac{TP+TN}{TP+T\ N+F\ N+FP}$
đúng khi dự đoán dương: độ chính xác = Nhớ lại: tần suất bộ phân loại đúng cho tất cả các trường $\frac{T\ P}{TP+FP}$
hợp tích cực: thu hồi = F-Score: phép đo đơn lẻ để mô tả hiệu suất: độ chính xác·gọi lại $\frac{T\ P}{TP+FN}$
F = 2 · độ chính xác + thu hồi

Đường cong ROC: biểu thị tỷ lệ dương thực và tỷ lệ dương giả cho các ngưỡng khác nhau hoặc trong đó mô hình xác định xem điểm dữ liệu là dương hay âm (ví dụ: nếu >0,8, hãy phân loại là dương). Vùng tốt nhất có thể có dưới đường cong ROC (AUC) là 1, trong khi ngẫu nhiên là 0,5 hoặc đường chéo chính.

Lỗi hồi quy
được định nghĩa là sự khác biệt giữa dự đoán y và kết quả thực tế y.
Sai số tuyệt đối: = y y Sai
số bình phương: 2 = (y y) i=1(yi ^{2}
Lỗi bình phương trung bình: MSE = Lỗi $\frac{1}{N}$ ^{2}
bình phương trung bình gốc: RMSD = √ Phân phối lỗi tuyệt đối MSE: Phân phối lỗi tuyệt đối trong biểu đồ: phải đối xứng, tập trung quanh 0, hình chuông và chứa các giá trị ngoại lệ cực kỳ hiếm.

Môi trường mô hình hóa-đánh giá

Số liệu đánh giá cung cấp công dụng cho các công cụ để ước tính lỗi, nhưng quy trình để có được ước tính tốt nhất là gì? Lấy mẫu lại liên quan đến việc liên tục lấy mẫu từ tập huấn luyện và điều chỉnh lại mô hình cho từng mẫu, điều này cung cấp cho chúng tôi thông tin bổ sung so với việc điều chỉnh mô hình một lần, chẳng hạn như thu được ước tính tốt hơn cho lỗi kiểm tra.

Ý chính
Dữ liệu đào tạo: dữ liệu được sử dụng để phù hợp với mô hình của bạn hoặc tập hợp được sử dụng để học
Dữ liệu xác thực: dữ liệu được sử dụng để điều chỉnh các tham số của mô hình
Dữ liệu thử nghiệm: dữ liệu được sử dụng để đánh giá mô hình của bạn tốt như thế nào. Lý tưởng nhất là mô hình của bạn không bao giờ chạm vào dữ liệu này cho đến khi kiểm tra/đánh giá lần cuối

Xác thực chéo Lớp các
phương pháp ước tính lỗi kiểm tra bằng cách giữ lại một tập hợp con dữ liệu huấn luyện từ quá trình khớp.
Bộ xác thực: chia dữ liệu thành tập huấn luyện và bộ xác thực. Huấn luyện mô hình về huấn luyện và ước tính lỗi kiểm tra bằng cách xác thực. ví dụ: CV nghỉ phép
một lần (LOOCV) chia theo tỷ lệ 80-20: chia dữ liệu thành tập huấn luyện và tập xác thực, nhưng tập xác thực bao gồm 1 quan sát. Sau đó lặp lại n-1 lần cho đến khi tất cả các quan sát được sử dụng làm xác nhận. Lỗi kiểm tra là
trung bình của n ước lượng lỗi kiểm tra này. k-
Fold CV: chia ngẫu nhiên dữ liệu thành k nhóm (gấp) có kích thước xấp xỉ bằng nhau. Phần đầu tiên được sử dụng để xác nhận và phần còn lại dùng để đào tạo. Sau đó lặp lại k lần và tìm giá trị trung bình của k ước lượng.

Phương pháp khởi
động dựa trên lấy mẫu ngẫu nhiên có thay thế.
Khởi động giúp định lượng độ không chắc chắn liên quan đến ước tính hoặc mô hình nhất định.

Khuếch đại các tập dữ liệu nhỏ Chúng
ta có thể làm gì khi không có đủ dữ liệu?
• Tạo các ví dụ tiêu cực- ví dụ: phân loại các ứng cử viên tổng thống, hầu hết mọi người sẽ không đủ tiêu chuẩn nên gắn nhãn hầu hết là không đủ tiêu
chuẩn • Dữ liệu tổng hợp- tạo dữ liệu bổ sung bằng cách thêm nhiễu vào dữ liệu thực

Hồi quy tuyến tính

Hồi quy tuyến tính là một công cụ đơn giản và hữu ích để dự đoán phản ứng định lượng. Mối quan hệ giữa biến đầu vào X = (X1, X2, ...Xp) và biến đầu ra Y có dạng:

$$Y \approx \beta 0 + \beta 1X1 + \ldots + \beta pXp +$$

β0...βp là các hệ số (tham số) chưa biết mà chúng ta đang cố gắng xác định. Các hệ số tốt nhất sẽ đưa chúng ta đến mức "phù hợp" tốt nhất, có thể tìm thấy bằng cách giảm thiểu bình phương tổng dư (RSS) hoặc tổng chênh lệch giữa giá trị thứ i thực tế và giá trị thứ i dự đoán. RSS = $\sum_{i=1}^{N}$ không, trong đó không = yi yˆi

Làm thế nào để tìm được sự phù hợp nhất?

Dạng ma trận: Chúng ta có thể giải phương trình dạng đóng cho vectơ hệ số w: w = (X là dữ liệu ^{T} X) 1X T Y . X đại diện đầu vào và Y đại diện cho dữ liệu đầu ra. Phương pháp này được sử dụng cho các ma trận nhỏ hơn, vì việc đảo ngược ma trận rất tốn kém về mặt tính toán.

Giảm dần độ dốc: Thuật toán tối ưu hóa bậc nhất.

Chúng ta có thể tìm giá trị cực tiểu của hàm lồi bằng cách bắt đầu tại một điểm tùy ý và lặp đi lặp lại thực hiện các bước theo hướng đi xuống, có thể tìm thấy điểm này bằng cách lấy hướng âm của gradient. Sau vài lần lặp, cuối cùng chúng ta sẽ hội tụ về mức tối thiểu.

Trong trường hợp của chúng tôi, mức tối thiểu tương ứng với các hệ số có sai số tối thiểu hoặc dòng phù hợp nhất. Tốc độ học α xác định kích thước của các bước chúng ta thực hiện theo hướng đi xuống.

Thuật toán giảm độ dốc theo hai chiều. Lặp lại cho đến khi hội tụ.

1. w_0^{t+1} t := w_0 một J(w0, w1)
2. w_1^{t+1} t := w_1 một $\frac{}{w0}$ J_{w1}(w0, w1)

Đối với các hàm không lồi, việc giảm độ dốc không còn đảm bảo một giải pháp tối ưu vì có thể có các giá trị cực tiểu cục bộ. Thay vào đó, chúng ta nên chạy thuật toán từ các điểm bắt đầu khác nhau và sử dụng cực tiểu cục bộ tốt nhất mà chúng ta tìm được cho lời giải.

Giảm dần độ dốc ngẫu nhiên: thay vì thực hiện một bước sau khi lấy mẫu toàn bộ tập huấn luyện, chúng tôi lấy ngẫu nhiên một loạt dữ liệu huấn luyện nhỏ để xác định bước tiếp theo. Tính toán hiệu quả hơn và có thể dẫn đến sự hội tụ nhanh hơn.

Hồi quy tuyến tính II

Cải thiện việc lựa chọn tập hợp con/ tính năng hồi quy tuyến tính: cách tiếp cận liên quan đến việc xác định một tập hợp con của các yếu tố dự đoán p mà chúng tôi tin là có liên quan tốt nhất đến phản hồi. Sau đó, chúng tôi điều chỉnh mô hình bằng cách sử dụng tập hợp các biến được rút gọn. • Thu hẹp/ chính quy hóa lựa chọn tập hợp con tốt nhất, tiến lên và lùi: tất cả các biến được sử dụng, nhưng các hệ số ước tính được thu nhỏ về 0 so với ước tính bình phương nhỏ nhất. λ đại diện cho tham số điều chỉnh - khi λ tăng, độ linh hoạt giảm phương sai giảm nhưng độ lệch tăng. Tham số điều chỉnh là chìa khóa trong việc xác định điểm phù hợp giữa mức dưới và mức quá khớp. Ngoài ra, trong khi Ridge luôn tạo ra một mô hình với p biến, Lasso có thể buộc các hệ số bằng 0. • Lasso (L1): RSS tối thiểu + λ • Ridge (L2):

RSS tối thiểu + λ $\sum_{pj=1}$ |βj | $\sum_{pj=1}$ b_j^2

Giảm kích thước: chiếu p yếu tố dự đoán vào không gian con M chiều, trong đó M < p. Điều này đạt được bằng cách tính toán M kết hợp tuyến tính khác nhau của các biến. Có thể sử dụng PCA.

Linh tinh: Loại bỏ các ngoại lệ, chia tỷ lệ đặc trưng, loại bỏ đa cộng tuyến (các biến tương quan)

Đánh giá lỗi tiêu chuẩn dư độ chính xác của mô hình (RSE): RSE = Nói chung, càng nhỏ càng tốt. 2 R : Thước đo mức độ phù hợp biểu thị tỷ lệ phương sai được giải thích hoặc độ biến thiên của Y có thể được giải thích bằng X. Nó có giá trị từ 0 đến 1. Nói chung càng cao càng tốt. R = 1 trong đó Tổng $\frac{1}{n-2}$RSS. ^{2} $\frac{RSS}{pigh}$, phương (TSS) = (yi y¯)

Đánh giá ước lượng hệ số Sai số chuẩn (SE) của các hệ số có thể được sử dụng để thực hiện kiểm định giả thuyết về các hệ số: H0: Không có mối quan hệ giữa X và Y, Ha: Tồn tại một số mối quan hệ. Giá trị p có thể thu được và có thể được hiểu như sau: giá trị p nhỏ biểu thị rằng tồn tại mối quan hệ lại giữa yếu tố dự đoán (X) và phản hồi (Y). Ngưỡng giá trị p điển hình là khoảng 5 hoặc 1%.

Hồi quy logistic

Hồi quy logistic được sử dụng để phân loại, trong đó biến phản hồi mang tính phân loại chứ không phải số.

Mô hình hoạt động bằng cách dự đoán xác suất Y thuộc một danh mục cụ thể bằng cách trước tiên khớp dữ liệu với mô hình hồi quy tuyến tính, sau đó được chuyển đến hàm logistic (bên dưới). Hàm logistic sẽ luôn tạo ra một đường cong hình chữ S, do đó, bất kể X là gì, chúng ta luôn có thể có được câu trả lời hợp lý (trong khoảng từ 0 đến 1). Nếu khả năng xác suất cao hơn một ngưỡng xác định trước (ví dụ: P(Có) > 0,5), thì mô hình sẽ dự đoán Có.

$$p(X) = \frac{\ldots\beta 0+\beta 1X1+\ldots+\beta pXp\ 1+e}{\beta 0+\beta 1X1+\ldots+\beta pXp}$$

Làm thế nào để tìm hệ số tốt nhất?

Khả năng tối đa: Các hệ số β0...βp chưa được biết và phải được ước tính từ dữ liệu huấn luyện. Chúng tôi tìm kiếm các ước tính cho β0...βp sao cho xác suất dự đoán ˆp(xi) của mỗi quan sát là một số gần bằng 1 nếu nó được quan sát trong một lớp nhất định và gần bằng 0 nếu ngược lại.

Điều này được thực hiện bằng cách tối đa hóa hàm khả năng:

$$l(\beta 0, \beta 1) = \prod_{\text{tôi:yi=1}} p(xi) \prod_{\text{tôi : yi =1}} (1 \quad p(xi))$$

Các vấn đề tiềm ẩn

Các lớp mất cân bằng: sự mất cân bằng giữa các lớp trong dữ liệu huấn luyện dẫn đến các bộ phân loại kém. Nó có thể dẫn đến nhiều kết quả dương tính giả và cũng dẫn đến ít dữ liệu huấn luyện. Các giải pháp bao gồm buộc phải cân bằng dữ liệu bằng cách loại bỏ các quan sát khỏi lớp lớn hơn, sao chép dữ liệu từ lớp nhỏ hơn hoặc cân nhắc kỹ các ví dụ huấn luyện đối với các phiên bản của lớp lớn hơn.

Phân loại nhiều lớp: bạn càng cố gắng dự đoán nhiều lớp thì bộ phân loại càng khó hoạt động hiệu quả. Có thể thực hiện được hồi quy logistic, nhưng một cách tiếp cận khác, chẳng hạn như Phân tích phân biệt tuyến tính (LDA), có thể chứng minh tốt hơn.

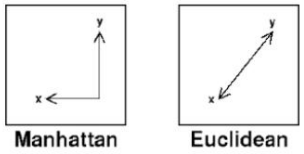
Phương pháp khoảng cách/mạng

giải thích các ví dụ dưới dạng các điểm trong không gian cung cấp một cách để tìm các nhóm hoặc cụm tự nhiên trong số dữ liệu, ví dụ: ngôi sao nào gần mặt trời của chúng ta nhất? Mạng cũng có thể được xây dựng từ các tập hợp điểm (đỉnh) bằng cách kết nối các điểm liên quan.

Đo khoảng cách/Đo độ tương tự Có một số cách đo khoảng cách giữa các điểm a và b trong chiều d - với khoảng cách gần hơn hàm ý sự giống nhau.

Số liệu khoảng cách Minkowski: $dk(a, b) = \sqrt[k]{\sum_{di=1} |ai \quad bi|^k}$ Tham số k cung cấp cách cân bằng giữa chênh lệch lớn nhất và tổng số thứ nguyên. Nói cách khác, giá trị lớn hơn của k nhấn mạnh hơn vào sự khác biệt lớn giữa các giá trị đặc trưng so với giá trị nhỏ hơn. Se-
chọn đúng k có thể tác động đáng kể đến ý nghĩa của hàm khoảng cách của bạn. Các giá trị phổ biến nhất là 1 và 2.

- Manhattan (k=1): khoảng cách khối thành phố hoặc tổng chênh lệch tuyệt đối giữa hai điểm
- Euclide (k=2): khoảng cách đường thẳng



Minkowski có trọng số: $dk(a, b) = \sqrt[k]{\sum_{di=1} wi|ai \quad bi|^k}$, trong một số trường hợp, không phải tất cả các thứ nguyên đều bằng nhau. Có thể truyền đạt ý tưởng này bằng cách sử dụng wi. Nói chung không phải là một ý tưởng hay - nên chuẩn hóa dữ liệu theo điểm Z trước khi tính toán khoảng cách.

Độ tương tự Cosine: $\cos(a, b) = \frac{a \cdot b}{|a||b|}$, tính toán độ tương tự giữa 2 vectơ khác 0, trong đó a · b là tích số chấm (chuẩn hóa từ 0 đến 1), giá trị cao hơn hàm ý nhiều vectơ giống nhau hơn

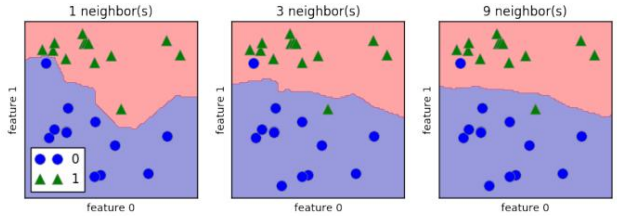
Phân kỳ Kullback-Leibler: $KL(A|B) = \text{Phân kỳ KL} \sum_{\text{của tôi}} ailog2 \frac{ai}{bi}$ đo khoảng cách giữa các phân bố xác suất bằng cách đo độ bất định thu được hoặc độ bất định bị mất khi thay thế phân phối A bằng phân phối B. Tuy nhiên, đây không phải là một thước đo mà là các dạng cơ sở cho Số liệu phân kỳ Jensen-Shannon.

Shannon: $JS(A, B) = \text{trong đó M là } \frac{1}{2}KL(A|M)+1\ 2KL(M|B)$, Jensen- trung bình của A và B. Hàm JS là thước đo phù hợp để tính khoảng cách giữa các xác suất sự phân phối

Phân loại hàng xóm gần nhất

Các hàm khoảng cách cho phép chúng ta xác định các điểm gần nhất với một mục tiêu nhất định hoặc các điểm lân cận gần nhất (NN) với một điểm nhất định. Ưu điểm của NN bao gồm tính đơn giản, khả năng diễn giải và tính phi tuyến tính.

k-Láng giềng gần nhất Cho một số nguyên dương k và một điểm x0, bộ phân loại KNN trước tiên xác định k điểm trong dữ liệu huấn luyện giống với x0 nhất, sau đó ước tính xác suất có điều kiện của x0 nằm trong lớp j là một phần của k điểm có giá trị thuộc về j. Giá trị tối ưu cho k có thể được tìm thấy bằng cách sử dụng xác thực chéo.



Thuật toán KNN

1. Tính khoảng cách D(a,b) từ điểm b đến tất cả các điểm 2. Chọn k điểm gần nhất và nhãn của chúng 3. Lớp đầu ra có nhãn thường xuyên nhất tính bằng k điểm Tối ưu hóa KNN So sánh điểm truy

vấn a trong d chiều với n tàu- các ví dụ tính toán với thời gian chạy là O(nd), điều này có thể gây ra độ trễ khi số điểm đạt tới hàng triệu hoặc hàng tỷ. Các lựa chọn phổ biến để tăng tốc KNN bao gồm: • Sơ đồ Vernoi: phân chia mặt phẳng thành các vùng dựa trên khoảng cách đến các điểm trong một tập hợp con cụ thể của mặt phẳng

- Chỉ mục lưới: chia không gian thành các hộp hoặc lưới d chiều và tính toán NN trong cùng ô với điểm • Băm nhạy cảm cục bộ (LSH):

từ bỏ ý tưởng tìm chính xác các lân cận gần nhất. Thay vào đó, hãy tập hợp các điểm lân cận để nhanh chóng tìm ra nhóm B thích hợp nhất cho điểm truy vấn của chúng ta. LSH được xác định bởi hàm băm h(p) lấy một điểm/vectơ làm đầu vào và tạo ra một số/mã làm đầu ra, sao cho có khả năng h(a) = h(b) nếu a và b gần nhau nhau và h(a)!= h(b) nếu chúng ở xa nhau.

Phân cụm

Phân cụm là vấn đề nhóm các điểm theo độ tương tự bằng cách sử dụng số liệu khoảng cách, phản ánh một cách lý tưởng những điểm tương đồng mà bạn đang tìm kiếm. Thông thường các mục đều đến từ các "nguồn" hợp lý và việc phân cụm là một cách hay để tiết lộ những nguồn gốc đó. Có lẽ điều đầu tiên cần làm với bất kỳ tập dữ liệu nào. Các ứng dụng có thể bao gồm: phát triển giả thuyết, mô hình hóa trên các tập hợp dữ liệu nhỏ hơn, giảm dữ liệu, phát hiện ngoại lệ.

Phân cụm K-Means Thuật toán đơn giản và tinh tế để phân vùng tập dữ liệu thành K cụm riêng biệt, không chồng chéo.

1. Chọn K. Gán ngẫu nhiên một số từ 1 đến K cho mỗi quan sát. Những thứ này đóng vai trò ban đầu bài tập cụm 2. Lặp lại cho đến khi bài tập cụm ngừng thay đổi
   (a) Đối với mỗi cụm K, hãy tính trọng tâm của cụm. Trọng tâm của cụm thứ k là vectơ của phương tiện đặc trưng p cho các quan sát trong cụm thứ k.
   (b) Chỉ định từng quan sát cho cụm có trọng tâm gần nhất (trong đó gần nhất được xác định bằng số liệu khoảng cách).

Vì kết quả của thuật toán phụ thuộc vào các phép gán ngẫu nhiên ban đầu, nên lặp lại thuật toán từ các lần khởi tạo ngẫu nhiên khác nhau để thu được kết quả tổng thể tốt nhất. Có thể sử dụng MSE để xác định phân công cụm nào tốt hơn.

Phân cụm theo thứ bậc Thuật toán phân cụm thay thế không yêu cầu chúng ta phải cam kết với một K cụ thể. Một ưu điểm khác là nó mang lại một hình ảnh trực quan đẹp mắt được gọi là chương trình dendro. Các quan sát hợp nhất ở phía dưới là tương tự nhau, trong đó các quan sát ở phía trên khá khác nhau - chúng tôi đưa ra kết luận dựa trên vị trí trên trục dọc thay vì trục ngang.

1. Bắt đầu với n quan sát và đo lường tất cả những khác biệt $\frac{(n)n \ 1}{2}$ theo cặp. Điều trị từng quan sát- tion như cụm riêng của nó.
2. Với i = n, n-1, ...2
   (a) Kiểm tra tất cả các điểm khác nhau giữa các cụm theo cặp giữa các cụm i và xác định cặp cụm ít khác nhau nhất (giống nhau nhất). Hợp nhất hai cụm này. Sự khác biệt giữa hai cụm này cho thấy chiều cao trong chương trình dendro nơi nên đặt sự hợp nhất. (b) Chỉ định từng quan sát cho cụm có trọng tâm gần nhất (trong đó gần nhất được xác định bằng số liệu khoảng cách).

Liên kết: Complete (độ khác biệt tối đa), Single (min), Av-erage, Centroid (giữa centroid của cụm A và B)

Học máy Phần I

So sánh sức mạnh và khả năng biểu đạt của thuật toán ML : Các phương pháp ML khác nhau về độ phức tạp. Hồi quy tuyến tính phù hợp với các hàm tuyến tính trong khi NN xác định ranh giới phân tách tuyến tính từng phần. Các mô hình phức tạp hơn có thể cung cấp các mô hình chính xác hơn nhưng có nguy cơ bị trang bị quá mức.
Khả năng diễn giải: một số mô hình minh bạch và dễ hiểu hơn các mô hình khác (mô hình hộp trắng so với mô hình hộp đen)

Dễ sử dụng: một số mô hình có ít tham số/quyết định (hồi quy tuyến tính/NN), trong khi các mô hình khác yêu cầu nhiều quyết định hơn để tối ưu hóa (SVM)
Tốc độ đào tạo: các mô hình khác nhau ở tốc độ chúng phù hợp với các tham số cần
thiết Tốc độ dự đoán: các mô hình khác nhau ở tốc độ đưa ra dự đoán cho một truy vấn

Method	Power of Expression	Ease of Interpretation	Ease of Use	Training Speed	Prediction Speed
Linear Regression	5	9	9	9	9
Nearest Neighbor	5	9	8	10	2
Naive Bayes	4	8	7	9	8
Decision Trees	8	8	7	7	9
Support Vector Machines	8	6	6	7	7
Boosting	9	6	6	6	6
Graphical Models	9	8	3	4	4
Deep Learning	10	3	4	3	7

Naive Bayes
Các phương pháp Naive Bayes là một tập hợp các thuật toán học có giám sát dựa trên việc áp dụng định lý Bayes với giả định "ngây thơ" về tính độc lập giữa mọi cặp đặc trưng.

Bài toán: Giả sử chúng ta cần phân loại vectơ X = x1...xn thành m lớp, C1...Cm. Chúng ta cần tính xác suất của từng lớp có thể cho X, để có thể gán cho X nhãn của lớp có xác suất cao nhất. Chúng ta có thể tính xác suất bằng Định lý Bayes:

$$P(Ci|X) = \frac{P(X|Ci)P(Ci)}{P(X)}$$

Ở đâu:

1. P(Ci): xác suất trước thuộc lớp i 2. P(X): hằng số chuẩn hóa hoặc xác suất nhìn thấy vectơ đầu vào đã cho trên tất cả các vectơ đầu vào có thể có 3. P(X|Ci): xác suất có điều kiện khi nhìn thấy vectơ đầu vào X đã cho, chúng ta biết lớp là Ci

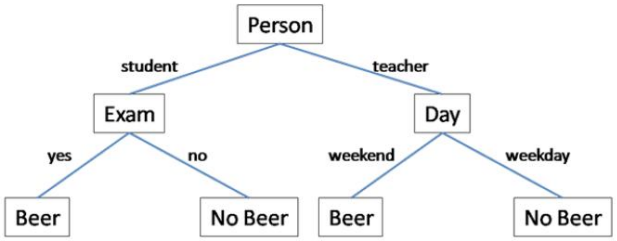
Mô hình dự đoán sẽ chính thức trông như sau:

$$C(X) = \text{argmaxi} \quad \text{classes(t)} \quad \frac{P(X|Ci)P(Ci)}{P(X)}$$

trong đó C(X) là dự đoán được trả về cho đầu vào X.

Học máy Phần II

Cây quyết định
Cấu trúc phân nhánh nhị phân được sử dụng để phân loại một vectơ đầu vào X tùy ý. Mỗi nút trong cây chứa một so sánh tính năng đơn giản với một số trường (xi > 42?).
Kết quả của mỗi phép so sánh là đúng hoặc sai, điều này quyết định xem chúng ta nên tiếp tục với nút con bên trái hay bên phải của nút đã cho. Đôi khi còn được gọi là cây phân loại và hồi quy (CART).



Ưu điểm: Không tuyến tính, hỗ trợ các biến phân loại, dễ diễn giải, ứng dụng vào hồi quy.
Nhược điểm: Dễ bị trang bị quá mức, không ổn định (không chịu được tiếng ồn), phương sai cao, độ lệch thấp

Lưu ý: hiếm khi có mô hình nào chỉ sử dụng một cây quyết định. Thay vào đó, chúng tôi tổng hợp nhiều cây quyết định bằng các phương pháp như tập hợp, đóng gói và tăng cường.

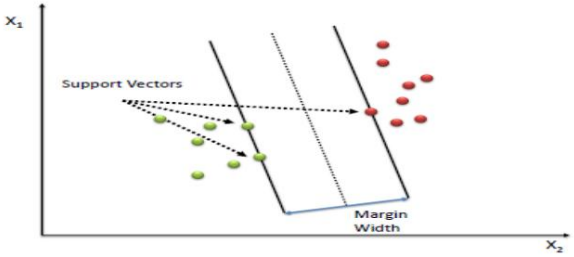
Ensembles, Bagging, Random Forests, Boosting Ensemble learning là chiến lược kết hợp nhiều phân loại/mô hình khác nhau thành một mô hình dự đoán. Nó xoay quanh ý tưởng bỏ phiếu: một cách tiếp cận được gọi là "sự khôn ngoan của đám đông". Lớp được dự đoán nhiều nhất sẽ là lớp dự đoán cuối cùng.

Đóng bao: phương pháp tổng hợp hoạt động bằng cách lấy các mẫu con B của dữ liệu huấn luyện và xây dựng cây B, mỗi cây huấn luyện trên một mẫu con riêng biệt dưới dạng Rừng ngẫu nhiên: xây dựng dựa trên việc đóng bao bằng cách giải mã các cây. Chúng tôi làm mọi thứ tương tự như trong việc đóng bao, nhưng khi chúng tôi xây dựng cây, mỗi khi chúng tôi xem xét việc phân tách, một mẫu ngẫu nhiên của các yếu tố dự đoán p sẽ được chọn làm các đối tượng có thể phân chia chứ không phải tập hợp đầy đủ (thường là m ≈ √p). Khi m = p thì chúng ta chỉ đang đóng bao.
Tăng cường: ý tưởng chính là cải thiện mô hình của chúng tôi khi nó hoạt động không tốt bằng cách sử dụng thông tin từ các bộ phân loại được xây dựng trước đó. Người học chậm. Có 3 tham số điều chỉnh: số lượng phân loại B, tham số học λ, độ sâu tương tác d (điều khiển thứ tự tương tác của mô hình).

Học máy Phần III

Máy vectơ hỗ trợ Hoạt động bằng cách xây dựng một siêu phẳng phân cách các điểm giữa hai lớp. Siêu phẳng được xác định bằng cách sử dụng siêu phẳng lề tối đa, là siêu phẳng có khoảng cách tối đa từ các quan sát huấn luyện.

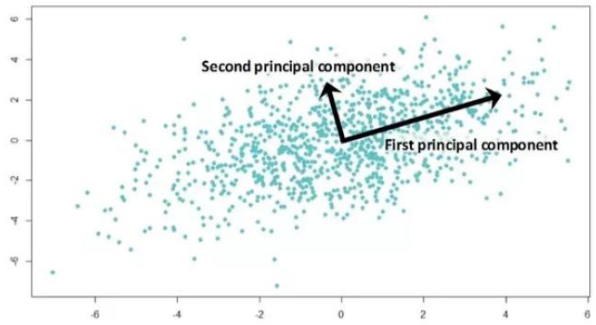
Khoảng cách này được gọi là lề. Các điểm rơi về một phía của siêu phẳng được phân loại là -1 và +1 còn lại.



Phân tích thành phần chính (PCA)
Các thành phần chính cho phép chúng ta tóm tắt một tập hợp các biến tương quan với một tập hợp các biến nhỏ hơn giải thích chung hầu hết các biến đổi trong tập hợp ban đầu. Về cơ bản, chúng tôi đang "bỏ" các biến tính năng ít quan trọng nhất.

Phân tích thành phần chính là quá trình tính toán các thành phần chính và sử dụng chúng để phân tích và hiểu dữ liệu. PCA là một cách tiếp cận không giám sát và được sử dụng để giảm kích thước, trích xuất tính năng và trực quan hóa dữ liệu.

Các biến sau khi thực hiện PCA là độc lập. Các biến mở rộng cũng rất quan trọng khi thực hiện PCA.



Học máy Phần IV

Thuật ngữ và khái niệm ML

Các tính năng: dữ liệu/biến đầu vào được sử dụng bởi mô hình ML Kỹ thuật tính năng: chuyển đổi các tính năng đầu vào để hữu ích hơn cho các mô hình. ví dụ: ánh xạ các danh mục vào các nhóm, chuẩn hóa từ -1 đến 1, loại bỏ null Train/Eval/Test: đào tạo là dữ liệu được sử dụng để tối ưu hóa mô hình, đánh giá được sử dụng để đánh giá mô hình trên dữ liệu mới trong quá trình đào tạo, kiểm tra được sử dụng để đưa ra kết quả cuối cùng kết quả Phân loại/Hồi quy: hồi quy là dự đoán một số (ví dụ: giá nhà đất), phân loại là dự đoán từ một tập hợp các danh mục (ví dụ: dự đoán màu đỏ/xanh dương/xanh lục) Hồi quy tuyến tính: dự đoán đầu ra bằng cách nhân và tính tổng các tính năng đầu vào với trọng số và độ lệch Hồi quy logistic: tương tự như hồi quy tuyến tính nhưng dự đoán xác suất Quá khớp: mô hình
hoạt động tốt trên dữ liệu đầu vào nhưng kém trên dữ liệu thử nghiệm (chiến đấu bằng cách bỏ học, dừng sớm- ping hoặc giảm số nút hoặc lớp)

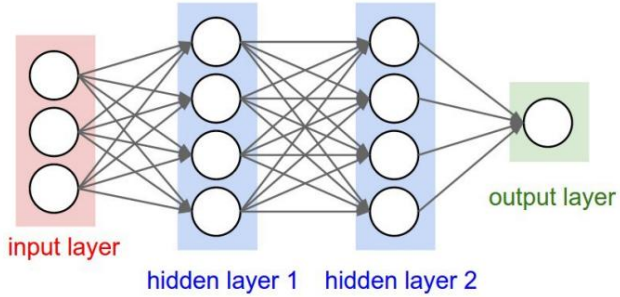
Xu hướng/Phương sai: mức độ đầu ra được xác định bởi các tính năng. nhiều phương sai hơn thường có nghĩa là trang bị quá mức, nhiều sai lệch hơn có thể có nghĩa là một mô hình xấu

Chính quy hóa: nhiều cách tiếp cận khác nhau để giảm tình trạng quá khớp, bao gồm thêm trọng số vào hàm mất mát, loại bỏ ngẫu nhiên các lớp (bỏ học)
Học tập theo nhóm: đào tạo nhiều mô hình với các tham số khác nhau để giải quyết cùng một vấn đề Thử nghiệm A/B: cách thống kê để so sánh 2+ kỹ thuật để xác định kỹ thuật nào hoạt động tốt hơn và cả liệu sự khác biệt có ý nghĩa thống kê hay không Mô hình cơ sở: mô hình/kinh nghiệm đơn giản được sử dụng làm điểm tham chiếu để so sánh mức độ hoạt động của một mô hình Xu hướng: thành kiến hoặc thiên vị đối với một số sự vật, con người hoặc nhóm hơn những thứ khác có thể ảnh hưởng đến việc thu thập/lấy mẫu và giải thích dữ liệu, thiết kế của hệ thống và cách người dùng tương
tác với một mô hình. hệ thống Mô hình động: mô hình được đào tạo trực tuyến theo
kiểu cập nhật liên tục Mô hình tĩnh: mô hình
được đào tạo ngoại tuyến Chuẩn hóa: quá trình chuyển đổi một phạm vi giá trị thực tế thành một phạm vi giá trị tiêu chuẩn, thường là -1 đến +1 một cách độc lập và giống hệt nhau Đã phân phối (iid): dữ liệu được rút ra từ một phân phối không thay đổi và trong đó mỗi giá trị được rút ra không phụ thuộc vào các giá trị được rút ra trước đó; lý tưởng nhưng hiếm thấy trong đời thực Siêu tham số: "nút" mà bạn điều chỉnh trong quá
trình đào tạo mô hình liên tiếp Tổng quát hóa: đề cập đến khả năng của mô hình trong việc đưa ra dự đoán chính xác về dữ liệu mới, chưa từng thấy trước đây,
trái ngược với dữ liệu đã sử dụng để huấn luyện mô hình Cross-Entropy: định lượng sự khác biệt giữa hai phân bố xác suất

Học sâu Phần I

Học sâu là gì?

Học sâu là một tập hợp con của học máy. Một kỹ thuật DL phổ biến dựa trên Mạng thần kinh (NN), mô phỏng một cách lỏng lẻo bộ não con người và các cấu trúc mã được sắp xếp theo lớp. Đầu vào của mỗi lớp là đầu ra của lớp trước, mang lại các tính năng cấp cao hơn dần dần và xác định hệ thống phân cấp. Mạng lưới thần kinh sâu chỉ là một NN có nhiều hơn 1 lớp ẩn.



Hãy nhớ lại rằng việc học thống kê là việc xấp xỉ f(X). Mạng lưới thần kinh được gọi là các công cụ mô phỏng xấp xỉ phổ quát, nghĩa là cho dù hàm có phức tạp đến đâu thì vẫn tồn tại một NN có thể (xấp xỉ) thực hiện công việc.
Chúng ta có thể tăng độ gần đúng (hoặc độ phức tạp) bằng cách thêm nhiều lớp và nơ-ron ẩn hơn.

Kiến trúc phổ biến

Có nhiều loại NN khác nhau phù hợp cho
một số vấn đề nhất định, phụ thuộc vào kiến trúc của NN.

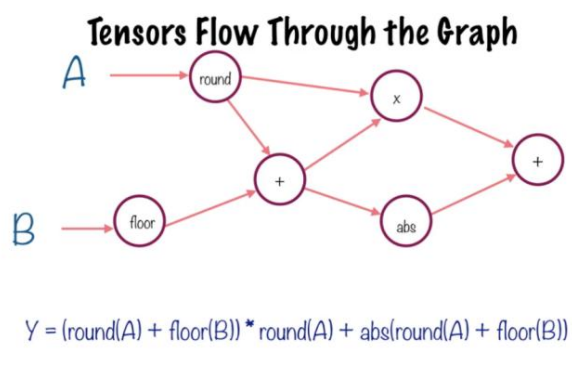
Trình phân loại tuyến tính: lấy các tính năng đầu vào và kết hợp chúng với các trọng số và độ lệch để dự đoán giá trị đầu ra DNN: mạng lưới thần kinh sâu, chứa các lớp nút trung gian đại diện cho "các tính năng ẩn" và các hàm kích hoạt để biểu thị tính phi tuyến tính CNN: NN
tích chập, có sự kết hợp của các lớp chập, gộp, dày đặc. phổ biến cho việc phân loại hình ảnh.

Học chuyển giao: sử dụng các mô hình đã được đào tạo hiện có làm điểm bắt đầu và thêm các lớp bổ sung cho trường hợp sử dụng cụ thể. Ý tưởng là các mô hình hiện có được đào tạo chuyên sâu biết các tính năng chung đóng vai trò là điểm khởi đầu tốt để đào tạo một mạng nhỏ trên các ví dụ cụ thể
RNN: NN lặp lại, được thiết kế để xử lý một chuỗi đầu vào có "bộ nhớ" của chuỗi. LSTM là một phiên bản ưa thích của RNN, phổ biến cho NLP GAN: NN đối nghịch chung, một
mô hình tạo các ví dụ giả và một mô hình khác được cung cấp cả ví dụ giả và ví dụ thực và được yêu cầu phân biệt Rộng và Sâu: kết hợp các bộ phân loại tuyến tính với độ
sâu các bộ phân loại mạng lưới thần kinh, các phần tuyến tính "rộng" biểu thị các ví dụ cụ thể giúp ghi nhớ và các phần "sâu" biểu thị các tính năng cấp cao khó hiểu

Học sâu Phần II

Dòng chảy căng

Tensorflow là một thư viện phần mềm nguồn mở để tính toán số bằng cách sử dụng biểu đồ luồng dữ liệu. Mọi thứ trong TF đều là một biểu đồ, trong đó các nút biểu thị các thao tác trên dữ liệu và các cạnh biểu thị dữ liệu. Giai đoạn 1 của TF đang xây dựng biểu đồ tính toán và giai đoạn 2 đang thực thi nó. Nó cũng được phân phối, nghĩa là nó có thể chạy trên một cụm máy hoặc chỉ một máy.
TF cực kỳ phổ biến/thích hợp để làm việc với Mạng thần kinh, vì cách TF thiết lập biểu đồ tính toán khá giống với NN.



Tenxơ

Trong đồ thị, tensor là các cạnh và là mảng dữ liệu đa chiều chảy qua đồ thị. Đơn vị dữ liệu trung tâm trong TF và bao gồm một tập hợp các giá trị nguyên thủy được định hình thành một mảng có số chiều bất kỳ.
Một tensor được đặc trưng bởi thứ hạng của nó (# chiều trong tensor), hình dạng (# chiều và kích thước của từng chiều), kiểu dữ liệu (kiểu dữ liệu của từng phần tử trong tensor)

Trình giữ chỗ và Biến : cách tốt

nhất để thể hiện trạng thái được chia sẻ, liên tục do chương trình của bạn thao tác. Đây là các tham số của mô hình ML được thay đổi/huấn luyện trong quá trình huấn luyện. Các biến đào tạo
Trình giữ chỗ: cách chỉ định đầu vào vào biểu đồ giữ vị trí cho Tensor sẽ được cung cấp khi chạy.
Chúng được chỉ định một lần, không thay đổi sau đó. Các nút đầu vào

Học sâu Phần III

Thuật ngữ và khái niệm Deep Learning

Neuron: nút trong NN, thường nhận nhiều giá trị đầu vào và tạo ra một giá trị đầu ra, tính toán giá trị đầu ra bằng cách áp dụng hàm kích hoạt (biến đổi tai không thẳng) cho tổng có trọng số của các giá trị đầu vào
Trọng số: các cạnh trong NN, mục tiêu của việc huấn luyện là xác định trọng số tối ưu cho từng tính năng; nếu trọng số = 0, tính năng tương ứng không đóng góp
Mạng nơ-ron: bao gồm các nơ-ron (các khối xây dựng đơn giản thực sự "học"), chứa các hàm kích hoạt giúp dự đoán các đầu ra phi tuyến tính
Hàm kích hoạt: các hàm toán học trong
giới thiệu tính phi tuyến tính cho mạng, ví dụ RELU, tanh
Sigmoid Function: hàm ánh xạ các số rất âm tới một số rất gần 0, các số lớn gần 1 và 0 đến 0,5. Hữu ích cho việc dự đoán xác suất Giảm dần/lan truyền ngược: các thuật toán tối ưu hóa tổn thất cơ bản, mà các trình tối ưu hóa khác thường dựa vào. Lan truyền ngược tương tự như giảm độ dốc nhưng đối với mạng lưới thần kinh

Trình tối ưu hóa: thao tác thay đổi trọng số và mã kép để giảm tổn thất, ví dụ Adagrad hoặc Adam Trọng số/Độ
lệch: trọng số là giá trị mà các tính năng đầu vào được nhân với nhau để dự đoán giá trị đầu ra. Độ lệch là giá trị của đầu ra có trọng số bằng 0.
Hội tụ: thuật toán hội tụ cuối cùng sẽ đạt được câu trả lời tối ưu, ngay cả khi rất chậm. Một thuật toán không hội tụ có thể không bao giờ đạt được câu trả lời tối ưu.
Tốc độ học tập: tốc độ mà trình tối ưu hóa thay đổi trọng số và độ lệch. Tốc độ học cao thường đào tạo nhanh hơn nhưng có nguy cơ không hội tụ, trong khi tốc độ thấp hơn đào tạo chậm hơn Tính không ổn định về mặt số học: các vấn đề với giá trị rất lớn/nhỏ do giới hạn số dấu phẩy động trong máy tính Nhúng:
ánh xạ từ các đối tượng rời rạc, chẳng hạn như từ, sang vectơ số thực. hữu ích vì các bộ phân loại/mạng nơ-ron hoạt động tốt trên các vectơ số thực Lớp tích chập: một loạt các phép toán tích chập, mỗi phép toán tác động lên một lát khác nhau của ma trận đầu vào Dropout: phương pháp chuẩn hóa trong huấn luyện NN, hoạt động bằng cách loại bỏ một lựa chọn ngẫu nhiên của một số đơn vị trong lớp mạng cho một bước gradient duy nhất Dừng sớm: phương pháp chuẩn hóa liên quan đến việc
kết thúc quá trình đào tạo mô hình sớm Giảm dần độ dốc: kỹ thuật giảm thiểu tổn thất bằng
cách tính toán độ dốc tổn thất theo các tham số của mô hình, dựa trên dữ liệu huấn luyện Pooling: Giảm ma trận (hoặc ma trận) được tạo bởi lớp tích chập trước đó
thành ma trận nhỏ hơn. Việc gộp chung thường liên quan đến việc lấy giá trị tối đa hoặc trung bình trên khu vực gộp

Dữ liệu lớn- Tổng quan về Hadoop

Dữ liệu không còn có thể vừa trong bộ nhớ trên một máy (nguyên khối), vì vậy một cách tính toán mới đã được phát minh bằng cách sử dụng một nhóm máy tính để xử lý "dữ liệu lớn" (phân tán) này. Một nhóm như vậy được gọi là cụm, tạo nên các cụm máy chủ. Tất cả các máy chủ này phải được phối hợp theo những cách sau: phân vùng dữ liệu, điều phối các tác vụ tính toán, xử lý khả năng chịu lỗi/khôi phục và phân bổ năng lực để xử lý.

Hadoop
Hadoop là một khung xử lý phân tán nguồn mở quản lý việc xử lý và lưu trữ dữ liệu cho các ứng dụng dữ liệu lớn chạy trong các hệ thống phân cụm. Nó bao gồm 3 thành phần chính: • Hệ thống tệp phân tán Hadoop (HDFS): một
hệ thống tệp phân tán cung cấp quyền truy cập thông lượng cao vào dữ liệu ứng dụng bằng cách phân vùng dữ liệu trên nhiều máy

• YARN: khung lập kế hoạch công việc và quản lý tài nguyên cụm (điều phối nhiệm vụ) • MapReduce: Hệ
thống dựa trên YARN để xử lý song song các tập dữ liệu lớn trên nhiều máy

HDFS
Mỗi đĩa trên một máy khác nhau trong một cụm bao gồm 1 nút chính và phần còn lại là nút công nhân/nút dữ liệu.
Nút chính quản lý hệ thống tệp tổng thể bằng cách lưu trữ cấu trúc thư mục và siêu dữ liệu của tệp. Các nút dữ liệu lưu trữ dữ liệu về mặt vật lý. Các tệp lớn được chia nhỏ và phân phối trên nhiều máy, chúng cũng được sao chép trên nhiều máy để cung cấp khả năng chịu lỗi.

Mô hình lập
trình song song MapReduce cho phép xử lý lượng dữ liệu khổng lồ bằng cách chạy các quy trình trên nhiều máy. Việc xác định công việc MapReduce yêu cầu hai giai đoạn: ánh xạ và thu nhỏ.
• Bản đồ: thao tác được
thực hiện song song trên các phần nhỏ của tập dữ liệu. đầu ra là cặp key-value < K, V > • Giảm: thao tác kết hợp các kết quả của Map

YARN- Yet Another Resource Negotiator Điều phối các
tác vụ đang chạy trên cụm và chỉ định các nút mới trong trường hợp thất bại. Bao gồm 2 thành phần phụ: trình quản lý tài nguyên và trình quản lý nút. Trình quản lý nguồn lại chạy trên một nút chính duy nhất và lập lịch các tác vụ trên các nút. Trình quản lý nút chạy trên tất cả các nút khác và quản lý các tác vụ trên nút riêng lẻ.

Dữ liệu lớn- Hệ sinh thái Hadoop

Toàn bộ hệ sinh thái các công cụ đã xuất hiện xung quanh Hadoop, dựa trên việc tương tác với HDFS.
Dưới đây là một số cái phổ biến:

Hive: phần mềm kho dữ liệu được xây dựng dựa trên Hadoop tạo điều kiện thuận lợi cho việc đọc, ghi và quản lý các tập dữ liệu lớn nằm trong bộ lưu trữ phân tán bằng các truy vấn giống SQL (HiveQL). Hive tóm tắt các công việc MapReduce cơ bản và trả về HDFS dưới dạng bảng (không phải HDFS).
Pig: ngôn ngữ kịch bản cấp cao (Pig Latin) cho phép viết các phép biến đổi dữ liệu phức tạp. Nó lấy dữ liệu không có cấu trúc/không đầy đủ từ các nguồn, làm sạch và đặt nó vào cơ sở dữ liệu/kho dữ liệu. Pig thực hiện ETL vào kho dữ liệu trong khi Hive truy vấn từ kho dữ liệu để thực hiện phân tích (GCP: DataFlow).
Spark: framework để viết các chương trình phân tán, nhanh chóng để xử lý và phân tích dữ liệu. Spark giải quyết các vấn đề tương tự như Hadoop MapReduce nhưng với cách tiếp cận nhanh trong bộ nhớ. Nó là một công cụ hợp nhất hỗ trợ các truy vấn SQL, truyền dữ liệu, học máy và xử lý đồ thị. Có thể hoạt động riêng biệt với Hadoop nhưng tích hợp tốt với Hadoop. Dữ liệu được xử lý bằng cách sử dụng Bộ dữ liệu phân phối linh hoạt (RDD), không thay đổi, được đánh giá một cách lười biếng và theo dõi dòng dõi. ; Hbase: hệ thống quản lý cơ sở dữ liệu hướng cột, không quan hệ, NoSQL chạy trên HDFS. Rất phù hợp với các tập dữ liệu thưa thớt (GCP: BigTable)

Flink/Kafka: khung xử lý luồng. Truyền phát hàng loạt dành cho các bộ dữ liệu hữu hạn, có giới hạn, được cập nhật định kỳ và xử lý chậm. Xử lý luồng dành cho các bộ dữ liệu không giới hạn, có cập nhật liên tục và xử lý ngay lập tức. Dữ liệu luồng và xử lý luồng phải được tách riêng thông qua hàng đợi tin nhắn.

Có thể nhóm dữ liệu phát trực tuyến (cửa sổ) bằng cách sử dụng cửa sổ giảm dần (thời gian không chồng chéo), trượt (thời gian chồng chéo) hoặc phiên (khoảng cách phiên).
Beam: mô hình lập trình để xác định và thực hiện các đường ống xử lý dữ liệu, bao gồm xử lý ETL, hàng loạt và luồng (liên tục). Sau khi xây dựng quy trình, nó được thực thi bởi một trong các back-end xử lý phân tán của Beam (Apache Apex, Apache Flink, Apache Spark và Google Cloud Dataflow). Được mô hình hóa dưới dạng đồ thị chu kỳ có hướng (DAG).

Oozie: hệ thống lập lịch công việc để quản lý các công việc của

Hadoop Sqoop: chuyển khung để chuyển lượng lớn dữ liệu vào HDFS từ cơ sở dữ liệu quan hệ (MySQL)

SQL Phần I

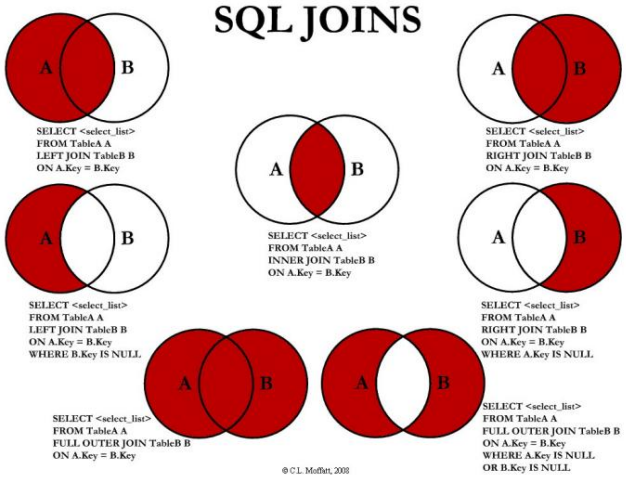
Ngôn ngữ truy vấn có cấu trúc (SQL) là ngôn ngữ khai báo được sử dụng để truy cập và thao tác dữ liệu trong cơ sở dữ liệu.
Thông thường cơ sở dữ liệu là Hệ thống quản lý cơ sở dữ liệu quan hệ (RDBMS), lưu trữ dữ liệu được sắp xếp trong các bảng cơ sở dữ liệu quan hệ. Một bảng được sắp xếp theo cột và hàng, trong đó các cột thể hiện đặc tính của dữ liệu được lưu trữ và các hàng thể hiện các mục nhập dữ liệu thực tế.

Truy vấn cơ bản -
lọc các cột: SELECT col1, col3... FROM table1 - lọc các hàng: WHERE col4 = 1 AND col5 = 2 - tổng hợp dữ liệu: GROUP BY. . . - giới hạn dữ liệu tổng hợp: CÓ số(*) > 1 - thứ tự của kết quả: ORDER BY col2

Từ khóa hữu ích cho SELECT
DISTINCT- trả về kết quả duy nhất
GIỮA a VÀ b- giới hạn phạm vi, các giá trị có thể là số, văn bản hoặc ngày
THÍCH- tìm kiếm mẫu trong văn bản cột
IN (a, b, c) - kiểm tra xem giá trị có nằm trong số đã cho không

Sửa đổi dữ liệu - cập nhật dữ liệu cụ thể với mệnh đề WHERE:
CẬP NHẬT bảng1 SET col1 = 1 WHERE col2 = 2 - chèn giá trị theo cách thủ công
CHÈN VÀO bảng1 (col1,col3) GIÁ TRỊ (val1,val3); - bằng cách sử dụng kết quả của một truy vấn
CHÈN VÀO bảng1 (col1,col3) CHỌN col,col2
Từ bảng2;

Tham gia Mệnh đề THAM GIA được sử dụng để kết hợp các hàng từ hai hoặc nhiều bảng, dựa trên một cột có liên quan giữa chúng.



Python- Cấu trúc dữ liệu

Cấu trúc dữ liệu là một cách lưu trữ và thao tác dữ liệu và mỗi cấu trúc dữ liệu đều có điểm mạnh và điểm yếu riêng. Kết hợp với các thuật toán, cấu trúc dữ liệu cho phép chúng ta giải quyết vấn đề một cách hiệu quả. Điều quan trọng là phải biết các loại cấu trúc dữ liệu chính mà bạn sẽ cần để giải quyết vấn đề một cách hiệu quả.

Danh sách: hoặc mảng, trình tự sắp xếp của các đối tượng, có thể thay đổi

```
>>> l = [42, 3.14, "xin chào","thế giới"]
```

Bộ dữ liệu: giống như danh sách, nhưng không thay đổi được

```
>>> t = (42, 3.14, "xin chào","thế giới")
```

Từ điển: bảng băm, cặp khóa-giá trị, chưa sắp xếp

```
>>> d = {"cuộc sống": 42, "pi": 3,14}
```

Tập hợp: chuỗi các phần tử duy nhất có thể thay đổi, không có thứ tự. Frozensets chỉ là những bộ bất biến

```
>>> s = set([42, 3.14, "hello","world"])
```

Bộ sưu tập Mô-đun deque:
hàng đợi hai đầu, tổng quát hóa ngăn xếp và hàng đợi; hỗ trợ nối thêm, nối thêmLeft, bật, xoay, v.v.

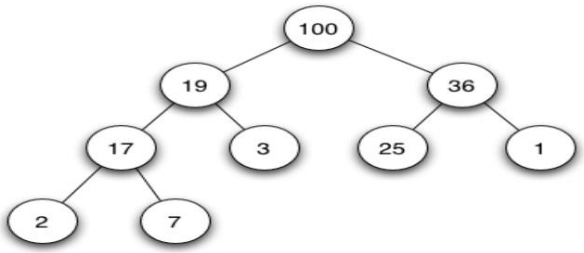
```
>>> s = deque([42, 3.14, "hello","world"])
```

Bộ đếm: lớp con dict, bộ sưu tập không có thứ tự trong đó các phần tử được lưu dưới dạng khóa và số đếm được lưu dưới dạng giá trị

```
>>> c = Bộ đếm('apple') >>>
print(c)
Bộ đếm({'p': 2, 'a': 1, 'l': 1, 'e': 1})
```

mô-đun heqpq
Hàng đợi Heap: hàng đợi ưu tiên, heap là cây nhị phân mà mỗi nút cha có giá trị lớn hơn hoặc bằng bất kỳ nút con nào của nó (max-heap), thứ tự rất quan trọng; hỗ trợ các cổng push, pop, pushpop, heapify, thay thế chức năng

```
>>> đống = []
>>> cho n trong dữ liệu:
...         heappush(đống, n) >>>
đống [0, 1,
3, 6, 2, 8, 4, 7, 9, 5]
```



Tài nguyên được đề xuất

- Sổ tay thiết kế khoa học dữ liệu (www.springer.com/us/book/9783319554433) • Giới thiệu về học thống kê (www-bcf.usc.edu/~gareth/ISL/) • Bảng tính xác suất (/www.wzchen.com/probability -cheatsheet/)

- Khóa học cấp tốc về máy học của Google (developers.google.com/machine-learning/crash-course/)