TÌM HIỂU VỀ TRÍ TUỆ NHÂN TẠO (AI) VÀ HỌC MÁY (ML)

Lê Bích Phượng

Trường Đại học Mỏ - Địa chất

lebichphuong@humg.edu.vn

Tháng 1 năm 2019

Mục lục

Lời mở đầu

Artificial intelligence hay trí tuệ nhân tạo và *Machine learning* gây nên cơn sốt công nghệ trên toàn thế giới trong vài năm nay. Trong giới học thuật, mỗi năm có hàng ngàn bài báo khoa học về đề tài này. Trong giới công nghiệp, từ các công ty lớn như Google, Facebook, Microsoft đến các công ty khởi nghiệp đều đầu tư vào machine learning. Hàng loạt các ứng dụng sử dụng machine learning ra đời trên mọi lĩnh vực của cuộc sống, từ khoa học máy tính đến những ngành ít liên quan hơn như vật lý, hóa học, y học, chính trị. *AlphaGo*, cỗ máy đánh cờ vây với khả năng tính toán trong một không gian có số lượng phần tử còn nhiều hơn số lượng hạt trong vũ trụ, tối ưu hơn bất kì đại kì thủ nào, là một trong rất nhiều ví dụ hùng hồn cho sự vượt trội của machine learning so với các phương pháp cổ điển.

Trong báo cáo học thuật này, tôi trình bày về hai khái niệm này và đưa ra một thí dụ minh họa.

Trí tuệ nhân tạo là gì?

Trí tuệ nhân tạo hay **trí thông minh nhân tạo** (tiếng Anh: *artificial intelligence* hay *machine intelligence*, thường được viết tắt là AI) là trí tuệ được biểu diễn bởi bất cứ một hệ thống nhân tạo nào. Thuật ngữ này thường dùng để nói đến các máy tính có mục đích không nhất định và ngành khoa học nghiên cứu về các lý thuyết và ứng dụng của trí tuệ nhân tạo.

Trí tuệ nhân tạo là gì?

Tuy rằng trí thông minh nhân tạo có nghĩa rộng như là trí thông minh trong các tác phẩm khoa học viễn tưởng, nó là một trong những ngành trọng yếu của tin học. Trí thông minh nhân tạo liên quan đến cách cư xử, sự học hỏi và khả năng thích ứng thông minh của máy móc. Các ví dụ ứng dụng bao gồm các tác vụ điều khiển, lập kế hoạch và lập lịch (*scheduling*), khả năng trả lời các câu hỏi về chẩn đoán bệnh, trả lời khách hàng về các sản phẩm của một công ty, nhận dạng chữ viết tay, nhận dạng tiếng nói và khuôn mặt. Bởi vậy, trí thông minh nhân tạo đã trở thành một môn học, với mục đích chính là cung cấp lời giải cho các vấn đề của cuộc sống thực tế. Ngày nay, các hệ thống nhân tạo được dùng thường xuyên trong kinh tế, y dược, các ngành kỹ thuật và quân sự, cũng như trong các phần mềm máy tính thông dụng trong gia đình và trò chơi điện tử.

Trí tuệ nhân tạo là gì?

Trí tuệ nhân tạo hay trí thông minh nhân tạo (tiếng Anh: Artificial intelligence hay tiếng Anh: Machine intelligence - AI) là một ngành thuộc lĩnh vực khoa học máy tính (tiếng Anh: Computer science). Là trí tuệ do con người lập trình tạo nên với mục tiêu giúp máy tính có thể tự động hóa các hành vi thông minh như con người. Trí tuệ nhân tạo khác với việc lập trình logic trong các ngôn ngữ lập trình là ở việc ứng dụng các hệ thống học máy (tiếng Anh: machine learning) để mô phỏng trí tuệ của con người trong các xử lý mà con người làm tốt hơn máy tính. Cụ thể, trí tuệ nhân tạo giúp máy tính có được những trí tuệ của con người như: biết suy nghĩ và lập luận để giải quyết vấn đề, biết giao tiếp do hiểu ngôn ngữ, tiếng nói, biết học và tự thích nghi,...

Trí tuệ nhân tạo là gì?

- Trí tuệ nhân tạo bao gồm các cơ sở lý thuyết và việc lập trình xây dựng của các hệ thống máy tính có thể thực hiện các nhiệm vụ thường đòi hỏi trí thông minh của con người như nhận thức thị giác, nhận dạng giọng nói, ra quyết định và dịch giữa các ngôn ngữ.
- Trí tuệ nhân tạo giúp tạo ra máy tính có khả năng suy nghĩ, máy tính có trí tuệ theo đầy đủ nghĩa của từ này (Haugeland, 1985).
- Trí tuệ nhân tạo là khoa học nghiên cứu xem thế nào để máy tính có thể thực hiện được những công việc mà con người làm tốt hơn máy tính (Rich và Knight,1991).

Trí tuệ nhân tạo là gì?

- Trí tuệ nhân tạo là khoa học nghiên cứu các hoạt động trí não thông qua các mô hình tính toán (Chaniaka và McDemott, 1985).
- Trí tuệ nhận tạo nghiên cứu các mô hình máy tính có thể nhận thức, lập luận và hành động (Winston, 1992).
- Trí tuệ nhân tạo nghiên cứu các hành vi thông minh mô phỏng các vật thể nhân tạo (Nilsson, 1998).
- Trí tuệ nhân tạo là khoa học nghiên cứu các hành vi thông minh nhằm giải quyết các vấn đề được đặt ra đối với các chương trình máy tính (Học viện Kỹ thuật Quân sự).

Mục tiêu của trí tuệ nhân tạo

Lý luận, giải quyết vấn đề

Các nhà nghiên cứu đầu tiên đã phát triển các thuật toán bắt chước theo lý luận từng bước mà con người sử dụng khi giải quyết các câu đố hoặc đưa ra các phương pháp loại trừ logic. Vào cuối những năm 1980 và 1990, nghiên cứu về AI đã phát triển các phương pháp xử lý thông tin không chắc chắn hoặc không đầy đủ, sử dụng các khái niệm từ xác suất và kinh tế.

Đối với những vấn đề khó, các thuật toán bắt buộc phải có phần cứng đủ mạnh để thực hiện phép tính toán khổng lồ - để trải qua "vụ nổ tổ hợp": lượng bộ nhớ và thời gian tính toán có thể trở nên vô tận nếu giải quyết một vấn đề khó. Mức độ ưu tiên cao nhất là tìm kiếm các thuật toán giải quyết vấn đề.

Mục tiêu của trí tuệ nhân tạo

Con người thường sử dụng các phán đoán nhanh và trực quan chứ không phải là phép khấu trừ từng bước mà các nghiên cứu AI ban đầu có thể mô phỏng. AI đã tiến triển bằng cách sử dụng cách giải quyết vấn đề "biểu tượng phụ": cách tiếp cận tác nhân được thể hiện nhấn mạnh tầm quan trọng của các kỹ năng cảm biến động đến lý luận cao hơn; nghiên cứu mạng thần kinh cố gắng để mô phỏng các cấu trúc bên trong não làm phát sinh kỹ năng này. Các phương pháp tiếp cận thống kê đối với AI bắt chước khả năng của con người.

Các trường phái trí tuệ nhân tạo

Trí tuệ nhân tạo (AI) chia thành hai trường phái tư duy: TTNT truyền thống và Trí tuệ tính toán.

TTNT truyền thống hầu như bao gồm các phương pháp hiện được phân loại là các phương pháp học máy (machine learning), đặc trưng bởi hệ hình thức (formalism) và phân tích thống kê. Nó còn được biết với các tên TTNT biểu tượng, TTNT logic, TTNT ngăn nắp (neat AI) và TTNT cổ điển (Good Old Fashioned Artificial Intelligence). (Xem thêm ngữ nghĩa học). Các phương pháp gồm có:

- Hệ chuyên gia: áp dụng các khả năng suy luận để đạt tới một kết luận. Một hệ chuyên gia có thể xử lý các lượng lớn thông tin đã biết và đưa ra các kết luận dựa trên các thông tin đó. Clippy chương trình trợ giúp có hình cái kẹp giấy của Microsoft Office là một ví dụ. Khi người dùng gõ phím, Clippy nhận ra các xu hướng nhất định và đưa ra các gợi ý.
- Lập luận theo tình huống.
- Mạng Bayes.

Các trường phái trí tuệ nhân tạo

Trí tuệ tính toán nghiên cứu việc học hoặc phát triển lặp (ví dụ: tinh chỉnh tham số trong hệ thống, chẳng hạn hệ thống connectionist). Việc học dựa trên dữ liệu kinh nghiệm và có quan hệ với Trí tuệ nhân tạo phi ký hiệu, TTNT lộn xộn (*scruffy AI*) và tính toán mềm (*soft computing*). Các phương pháp chính gồm có:

- Mạng neural: các hệ thống mạnh về nhận dạng mẫu (*pattern recognition*).
- Hệ mờ (*Fuzzy system*): các kỹ thuật suy luận không chắc chắn, đã được sử dụng rộng rãi trong các hệ thống công nghiệp hiện đại và các hệ thống quản lý sản phẩm tiêu dùng.
- Tính toán tiến hóa (*Evolutionary computation*): ứng dụng các khái niệm sinh học như quần thể, biến dị và đấu tranh sinh tồn để sinh các lời giải ngày càng tốt hơn cho bài toán. Các phương pháp này thường được chia thành các thuật toán tiến hóa (ví dụ thuật toán gene) và trí tuệ bầy đàn (*swarm intelligence*) (chẳng hạn hệ kiến).
- TTNT dựa hành vi (*Behavior based AI*): một phương pháp module để xây dựng các hệ thống TTNT bằng tay.

Các trường phái trí tuệ nhân tạo

Người ta đã nghiên cứu các hệ thống thông minh lai (*hybrid intelligent system*), trong đó kết hợp hai trường phái này. Các luật suy diễn của hệ chuyên gia có thể được sinh bởi mạng neural hoặc các luật dẫn xuất (*production rule*) từ việc học theo thống kê như trong kiến trúc ACT-R.

Các phương pháp trí tuệ nhân tạo thường được dùng trong các công trình nghiên cứu khoa học nhận thức (*cognitive science*), một ngành cố gắng tạo ra mô hình nhận thức của con người (việc này khác với các nghiên cứu TTNT, vì TTNT chỉ muốn tạo ra máy móc thực dụng, không phải tạo ra mô hình về hoạt động của bộ óc con người).

Machine learning là gì?

Machine learning

1. Machine learning và **trí tuệ nhân tạo** (Artificial Intelligence hay AI)
2. Machine learning và **Big Data**.
3. Machine learning và **dự đoán tương lai**.

Machine learning là gì?

Trí tuệ nhân tạo, AI, một cụm từ vừa gần gũi vừa xa lạ đối với chúng ta. Gần gũi bởi vì thế giới đang phát sốt với những công nghệ được dán nhãn AI. Xa lạ bởi vì một AI thực thụ vẫn còn nằm ngoài tầm với của chúng ta. Nói đến AI, hẳn mỗi người sẽ liên tưởng đến một hình ảnh khác nhau. Vài thập niên gần đây có một sự thay đổi về diện mạo của AI trong các bộ phim quốc tế. Trước đây, các nhà sản xuất phim thường xuyên đưa hình ảnh robot vào phim (như *Terminator*), nhằm gieo vào đầu người xem suy nghĩ rằng trí tuệ nhân tạo là một phương thức nhân bản con người bằng máy móc. Tuy nhiên, trong những bộ phim gần hơn về đề tài này, ví dụ như *Transcendence* do Johny Depp vào vai chính, ta không thấy hình ảnh của một con robot nào cả. Thay vào đó là một bộ não điện toán khổng lồ chỉ huy hàng vạn con Nanobot, được gọi là Singularity. Tất nhiên cả hai hình ảnh đều là hư cấu và giả tưởng, nhưng sự thay đổi như vậy cũng một phần nào phản ánh sự thay đổi ý niệm của con người về AI. AI bây giờ được xem như vô hình vô dạng, hay nói cách khác có thể mang bất cứ hình dạng nào. Vì nói về AI là nói về một *bộ não*, chứ không phải nói về một cơ thể, là software chứ không phải là hardware.

Machine learning là gì?

Trong giới hàn lâm, theo hiểu biết chung, AI là một ngành khoa học được sinh ra với mục đích làm cho máy tính có được trí thông minh. Mục tiêu này vẫn khá mơ hồ vì không phải ai cũng đồng ý với một định nghĩa thống nhất về trí thông minh. Các nhà khoa học phải định nghĩa một số mục tiêu cụ thể hơn, một trong số đó là việc làm cho máy tính lừa được *Turing Test*. Turing Test được tạo ra bởi Alan Turing (1912-1954), người được xem là cha đẻ của ngành khoa học máy tính hiện đại, nhằm phân biệt xem người đối diện có phải là người hay không.

Machine learning là gì?

AI thể hiện một *mục tiêu* của con người. Machine learning là một *phương tiện* được kỳ vọng sẽ giúp con người đạt được mục tiêu đó. Và thực tế thì machine learning đã mang nhân loại đi rất xa trên quãng đường chinh phục AI. Nhưng vẫn còn một quãng đường xa hơn rất nhiều cần phải đi. Machine learning và AI có mối quan hệ chặt chẽ với nhau nhưng không hẳn là trùng khớp vì một bên là mục tiêu (AI), một bên là phương tiện (machine learning). Chinh phục AI mặc dù vẫn là mục đích tối thượng của machine learning, nhưng hiện tại machine learning tập trung vào những mục tiêu ngắn hạn hơn như:

Làm cho máy tính có những khả năng nhận thức cơ bản của con người như nghe, nhìn, hiểu được ngôn ngữ, giải toán, lập trình, ...Hỗ trợ con người trong việc xử lý một khối lượng thông tin khổng lồ mà chúng ta phải đối mặt hàng ngày, hay còn gọi là Big Data.

Machine learning là gì?

Big Data thực chất không phải là một ngành khoa học chính thống. Đó là một cụm từ dân gian và được giới truyền thông tung hô để ám chỉ thời kì bùng nổ của dữ liệu hiện nay. Nó cũng không khác gì với những cụm từ như "cách mạng công nghiệp", "kỉ nguyên phần mềm". Big Data là một hệ quả tất yếu của việc mạng Internet ngày càng có nhiều kết nối. Với sự ra đời của các mạng xã hội nhưng Facebook, Instagram, Twitter, nhu cầu chia sẻ thông của con người tăng trưởng một cách chóng mặt. Youtube cũng có thể được xem là một mạng xã hội, nơi mọi người chia sẻ video và comment về nội dung của video.

Machine learning là gì?

Để hiểu được quy mô của Big Data, hãy xem qua những con số sau đây (tính đến thời điểm bài viết):

1. Khoảng *300 giờ video* được upload trên youtube trong mỗi phút (theo https://www.youtube.com/yt/press/statistics.html)
2. Hơn *900 triệu người* thật sự sử dụng Facebook mỗi ngày, 82.8% trong số đó ở ngoài Mỹ và Canada (theo http://newsroom.fb.com/company-info/)
3. Nhu cầu chia sẻ tăng đi đôi với việc nhu cầu tìm kiếm thông tin cũng tăng. Google phải xử lý *100 tỉ lượt* tìm kiếm mỗi tháng, tức là *3,3 tỉ lượt* mỗi ngày và *38.000 lượt* mỗi giây (theo http://www.internetlivestats.com/google-search-statistics/).

Và những con số này đang tăng lên theo từng giây!

Machine learning là gì?

Bùng nổ thông tin không phải là lý do duy nhất dẫn đến sự ra đời của cụm từ Big Data. Nên nhớ rằng Big Data xuất hiện mới từ vài năm gần đây nhưng khối lượng dữ liệu tích tụ kể từ khi mạng Internet xuất hiện vào cuối thế kỉ trước cũng không phải là nhỏ. Thế nhưng, lúc ấy con người ngồi quanh một đống dữ liệu và không biết làm gì với chúng ngoài lưu trữ và sao chép. Cho đến một ngày, các nhà khoa học nhận ra rằng trong đống dữ liệu ấy thực ra chứa một khối lượng tri thức khổng lồ. Những tri thức ấy có thể giúp cho ta hiểu thêm về con người và xã hội. Từ danh sách bộ phim yêu thích của một cá nhân chúng ta có thể rút ra được sở thích của người đó và giới thiệu những bộ phim người ấy chưa từng xem, nhưng phù hợp với sở thích. Từ danh sách tìm kiếm của cộng đồng mạng chúng ta sẽ biết được vấn đề nóng hổi nhất đang được quan tâm và sẽ tập trung đăng tải nhiều tin tức hơn về vấn đề đó. Big Data chỉ thực sự bắt đầu từ khi chúng ta hiểu được giá trị của thông tin ẩn chứa trong dữ liệu, và có đủ tài nguyên cũng như công nghệ để có thể khai thác chúng trên quy mô khổng lồ. Và không có gì ngạc nhiên khi machine learning chính là thành phần mấu chốt của công nghệ đó.

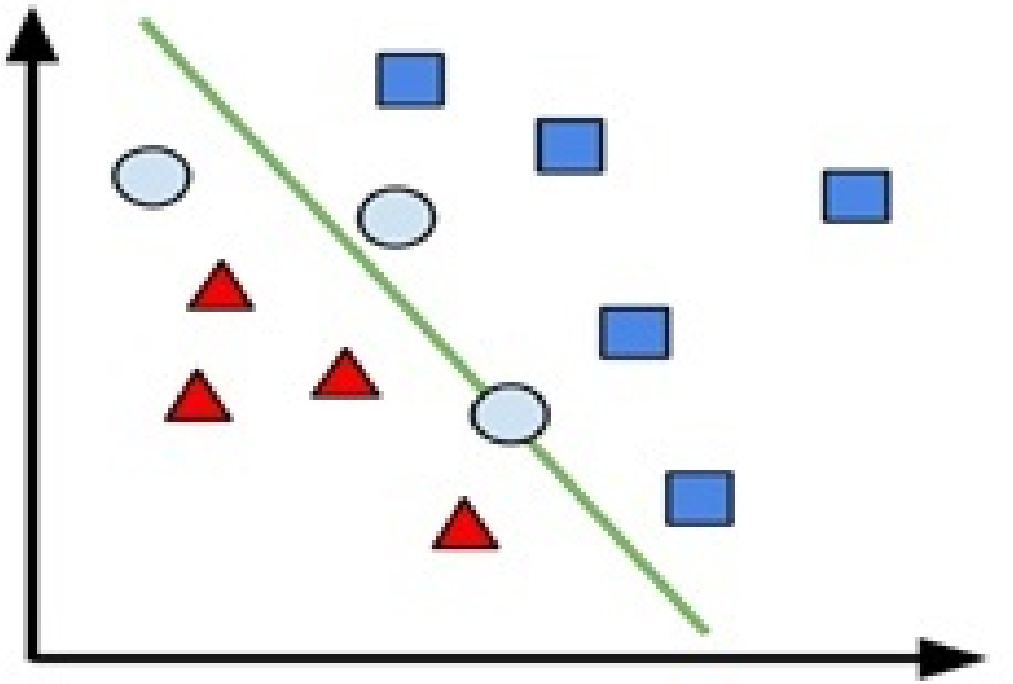
Machine learning là gì?

Ở đây ta có một quan hệ hỗ tương giữa machine Learning và Big Data: machine learning phát triển hơn nhờ sự gia tăng của khối lượng dữ liệu của Big Data; ngược lại, giá trị của Big Data phụ thuộc vào khả năng khai thác tri thức từ dữ liệu của machine learning.

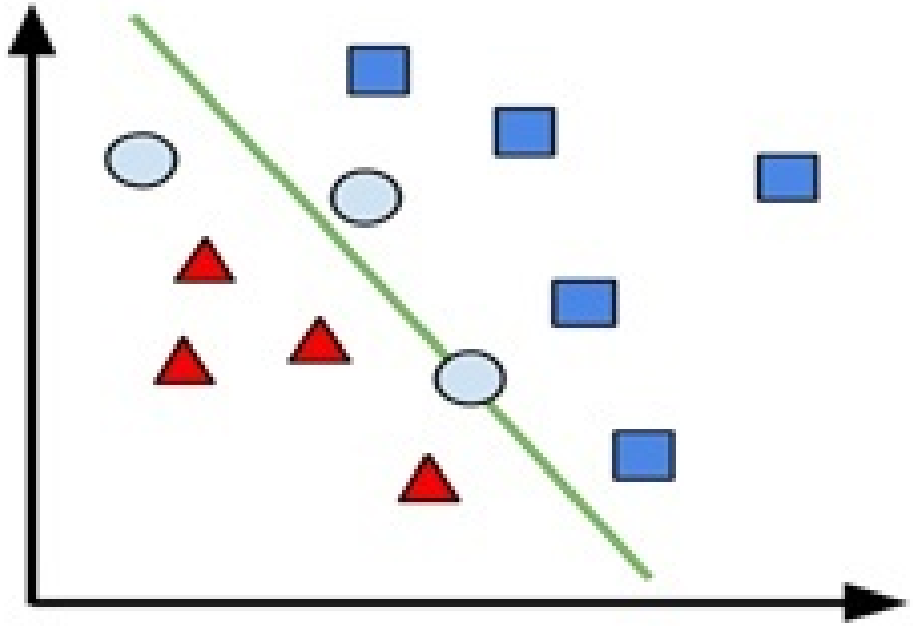
Ngược dòng lịch sử, machine learning đã xuất hiện từ rất lâu trước khi mạng Internet ra đời. Một trong những thuật toán machine learning đầu tiên là *thuật toán perceptron* được phát minh ra bởi Frank Rosenblatt vào năm 1957. Đây là một thuật toán kinh điển dùng để phân loại hai khái niệm.

Machine learning là gì?

Một ví dụ đơn giản là phân loại thư rác (tam giác) và thư bình thường (vuông). Chắc các bạn sẽ khó hình dung ra được làm thế nào để làm được điều đó. Đối với perceptron, điều này không khác gì với việc vẽ một đường thẳng trên mặt phẳng để phân chia hai tập điểm:



Machine learning là gì?



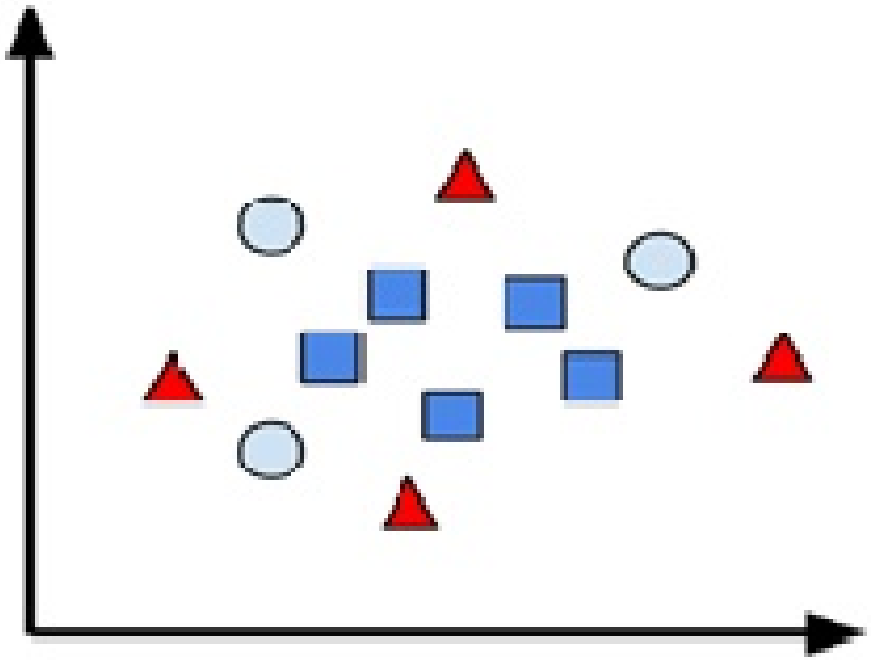
Những điểm tam giác và vuông đại diện cho những email chúng ta đã biết nhãn trước. Chúng được dùng để "huấn luyện" (train) perceptron. Sau khi vẽ đường thẳng chia hai tập điểm, ta nhận thêm các điểm chưa được dán nhãn, đại diện cho các email cần được phân loại (điểm tròn). Ta dán nhãn của một điểm theo nhãn của các điểm cùng nửa mặt phẳng với điểm đó.

Machine learning là gì?

Sơ lược quy trình phân loại thư được mô tả sau. Trước hết, ta cần một thuật toán để chuyển email thành những điểm dữ liệu. Công đoạn này rất rất quan trọng vì nếu chúng ta chọn được biểu diễn phù hợp, công việc của perceptron sẽ nhẹ nhàng hơn rất nhiều. Tiếp theo, perceptron sẽ đọc tọa độ của từng điểm và sử dụng thông tin này để cập nhật tham số của đường thẳng cần tìm. Các bạn có thể xem qua demo của perceptron (điểm xanh lá cây là điểm perceptron đang xử lý):

Machine learning là gì?

Vì là một thuật toán khá đơn giản, có rất nhiều vấn đề có thể nảy sinh với perceptron, ví dụ như điểm cần phân loại nằm ngay trên đường thẳng phân chia. Hoặc tệ hơn là với một tập dữ liệu phức tạp hơn, đường thẳng phân chia không tồn tại:



Lúc này, ta cần các loại đường phân chia "không thẳng". Nhưng đó lại là một câu chuyện khác.

Machine learning là gì?

Perceptron là một thuật toán **supervised learning**: ta đưa cho máy tính hàng loạt các ví dụ cùng câu trả lời mẫu với hy vọng máy tính sẽ tìm được những đặc điểm cần thiết để đưa ra dự đoán cho những ví dụ khác chưa có câu trả lời trong tương lai. Ngoài ra, cũng có những thuật toán machine learning không cần câu trả lời mẫu, được gọi là **unsupervised learning**. Trong trường hợp này, máy tính cố gắng khai thác ra cấu trúc ẩn của một tập dữ liệu mà không cần câu trả lời mẫu. Một loại machine learning khác được gọi là **reinforcement learning**. Trong dạng này, cũng không hề có câu trả lời mẫu, nhưng thay vì đó máy tính nhận được phản hồi cho mỗi hành động. Dựa vào phản hồi tích cực hay tiêu cực mà máy tính sẽ điều chỉnh hoạt động cho phù hợp.

Machine learning là gì?

Ví dụ

Mục tiêu của chiếc xe là leo lên được đỉnh đồi và lấy được ngôi sao. Chiếc xe có hai chuyển động tới và lui. Bằng cách thử các chuyển động và nhận được phản hồi là độ cao đạt được và thời gian để lấy được ngôi sao, chiếc xe dần trở nên thuần thục hơn trong việc leo đồi lấy sao.

Machine learning là gì?

Machine learning có mối quan hệ rất mật thiết đối với **statistics** (thống kê). Machine learning sử dụng các mô hình thống kê để "ghi nhớ" lại sự phân bố của dữ liệu. Tuy nhiên, không đơn thuần là ghi nhớ, machine learning phải có **khả năng tổng quát hóa** những gì đã được nhìn thấy và đưa ra dự đoán cho những trường hợp chưa được nhìn thấy. Bạn có thể hình dung một mô hình machine learning không có khả năng tổng quát như một đứa trẻ học vẹt: chỉ trả lời được những câu trả lời mà nó đã học thuộc lòng đáp án. Khả năng tổng quát là một khả năng tự nhiên và kì diệu của con người: bạn không thể nhìn thấy tất cả các khuôn mặt người trên thế giới nhưng bạn có thể nhận biết được một thứ có phải là khuôn mặt người hay không với xác suất đúng gần như tuyệt đối. Đỉnh cao của machine learning sẽ là mô phỏng được khả năng tổng quát hóa và suy luận này của con người.

Machine learning là gì?

Ví dụ

Giả sử bạn được đưa cho một đồng xu, rồi được yêu cầu tung đồng xu một số lần. Vấn đề đặt ra là: dựa vào những lần tung đồng xu đó, bạn hãy tiên đoán ra kết quả lần tung tiếp theo.

Machine learning là gì?

Ví dụ

Chỉ cần dựa vào tỉ lệ sấp/ngửa của những lần tung trước đó, bạn có thể đưa ra một dự đoán khá tốt. Nhưng nếu mỗi lần tung, người ta đưa cho bạn một đồng xu khác nhau thì mọi chuyện sẽ hoàn toàn khác. Các đồng xu khác nhau có xác suất sấp ngửa khác nhau. Lúc này việc dự đoán gần như không thể vì xác suất sấp ngửa của lần tung sau không hề liên quan gì đến lần tung trước. Điều tương tự cũng xảy ra với việc dự đoán tương lai bằng machine learning, nếu ta xem như mỗi ngày có một "đồng xu" được tung ra để xem một sự kiện có diễn ra hay không. Nếu "đồng xu" của ngày mai được chọn một cách tùy ý không theo phân bố nào cả thì machine learning sẽ thất bại. Rất may là trong nhiều trường hợp điều đó không hoàn toàn đúng, thế giới hoạt động theo những quy luật nhất định và machine learning có thể nhận ra những quy luật đó. Nhưng nói cho cùng, machine learning hoàn toàn không phải là một bà phủ thủy với quả cầu tiên tri mà cũng giống như chúng ta: phán đoán bằng cách tổng quát hóa những kinh nghiệm, những gì đã được học từ dữ liệu.

Phân nhóm các thuật toán của Machine Learning

Phân nhóm dựa trên phương thức học

Theo phương thức học, các thuật toán Machine Learning thường được chia làm 4 nhóm:

1. Supervise learning,
2. Unsupervised learning,
3. Semi-supervised lerning
4. Reinforcement learning.

Có một số cách phân nhóm không có Semi-supervised learning hoặc Reinforcement learning.

Phân nhóm các thuật toán của Machine Learning

Supervised Learning (Học có giám sát)

Supervised learning là thuật toán dự đoán đầu ra (outcome) của một dữ liệu mới (new input) dựa trên các cặp (*input, outcome*) đã biết từ trước. Cặp dữ liệu này còn được gọi là (*data, label*), tức (*dữ liệu, nhãn*). Supervised learning là nhóm phổ biến nhất trong các thuật toán Machine Learning.

Một cách toán học, Supervised learning là khi chúng ra có một tập hợp biến đầu vào $X = \{x_1, x_2, ..., x_N\}$ và một tập hợp nhãn tương ứng $Y = \{y_1, y_2, ..., y_N\}$, trong đó x_i, y_i là các vector. Các cặp dữ liệu biết trước $(x_i, y_i) \in X \times Y (x_i, y_i) \in X \times Y$ được gọi là tập *training data* (dữ liệu huấn luyện). Từ tập traing data này, chúng ta cần tạo ra một hàm số ánh xạ mỗi phần tử từ tập X sang một phần tử (xấp xỉ) tương ứng của tập Y:

$$y_i \approx f(x_i), \forall i = 1, 2, ..., N.$$

Mục đích là xấp xỉ hàm số ff thật tốt để khi có một dữ liệu xx mới, chúng ta có thể tính được nhãn tương ứng của nó $y = f(x)$.

Phân nhóm các thuật toán của Machine Learning

Ví dụ (1)

Trong nhận dạng chữ viết tay, ta có ảnh của hàng nghìn ví dụ của mỗi chữ số được viết bởi nhiều người khác nhau. Chúng ta đưa các bức ảnh này vào trong một thuật toán và chỉ cho nó biết mỗi bức ảnh tương ứng với chữ số nào. Sau khi thuật toán tạo ra một mô hình, tức một hàm số mà đầu vào là một bức ảnh và đầu ra là một chữ số, khi nhận được một bức ảnh mới mà mô hình **chưa nhìn thấy bao giờ**, *nó sẽ dự đoán bức ảnh đó chứa chữ số nào.*

Phân nhóm các thuật toán của Machine Learning

Ví dụ (1)

MNIST: bộ cơ sở dữ liệu của chữ số viết tay. (Nguồn: *Simple Neural Network implementation in Ruby)*

Phân nhóm các thuật toán của Machine Learning

Ví dụ này khá giống với cách học của con người khi còn nhỏ. Ta đưa bảng chữ cái cho một đứa trẻ và chỉ cho chúng đây là chữ A, đây là chữ B. Sau một vài lần được dạy thì trẻ có thể nhận biết được đâu là chữ A, đâu là chữ B trong một cuốn sách mà chúng chưa nhìn thấy bao giờ.

Phân nhóm các thuật toán của Machine Learning

Ví dụ (2)

Thuật toán dò các khuôn mặt trong một bức ảnh đã được phát triển từ rất lâu. Thời gian đầu, facebook sử dụng thuật toán này để chỉ ra các khuôn mặt trong một bức ảnh và yêu cầu người dùng tag friends - tức gán nhãn cho mỗi khuôn mặt. Số lượng cặp dữ liệu (khuôn mặt, tên người) càng lớn, độ chính xác ở những lần tự động tag tiếp theo sẽ càng lớn.

Ví dụ (3)

Bản thân thuật toán dò tìm các khuôn mặt trong 1 bức ảnh cũng là một thuật toán Supervised learning với training data (dữ liệu học) là hàng ngàn cặp (ảnh, mặt người) và (ảnh, không phải mặt người) được đưa vào. Chú ý là dữ liệu này chỉ phân biệt mặt người và không phải mặt người mà không phân biệt khuôn mặt của những người khác nhau.

Phân nhóm các thuật toán của Machine Learning

Thuật toán supervised learning còn được tiếp tục chia nhỏ ra thành hai loại chính:

1. Classification (Phân loại)
   Một bài toán được gọi là *classification* nếu các *label* của *input data* được chia thành một số hữu hạn nhóm. Ví dụ: Gmail xác định xem một email có phải là spam hay không; các hãng tín dụng xác định xem một khách hàng có khả năng thanh toán nợ hay không. Ba ví dụ phía trên được chia vào loại này.
2. Regression (Hồi quy)
   (tiếng Việt dịch là *Hồi quy*, tôi không thích cách dịch này vì bản thân không hiểu nó nghĩa là gì).
   Nếu *label* không được chia thành các nhóm mà là một giá trị thực cụ thể. Ví dụ: một căn nhà rộng $x\,\mathrm{m}^2$, có y phòng ngủ và cách trung tâm thành phố $z\,\mathrm{km}$ sẽ có giá là bao nhiêu?

Phân nhóm các thuật toán của Machine Learning

Gần đây Microsoft có một ứng dụng dự đoán giới tính và tuổi dựa trên khuôn mặt. Phần dự đoán giới tính có thể coi là thuật toán **Classification**, phần dự đoán tuổi có thể coi là thuật toán **Regression**. *Chú ý rằng phần dự đoán tuổi cũng có thể coi là* **Classification** *nếu ta coi tuổi là một số nguyên dương không lớn hơn 150, chúng ta sẽ có 150 class (lớp) khác nhau.*

Phân nhóm các thuật toán của Machine Learning

Unsupervised Learning (Học không giám sát)

Trong thuật toán này, chúng ta không biết được *outcome* hay *nhãn* mà chỉ có dữ liệu đầu vào. Thuật toán unsupervised learning sẽ dựa vào cấu trúc của dữ liệu để thực hiện một công việc nào đó, ví dụ như phân nhóm (clustering) hoặc giảm số chiều của dữ liệu (dimension reduction) để thuận tiện trong việc lưu trữ và tính toán.

Một cách toán học, Unsupervised learning là khi chúng ta chỉ có dữ liệu vào X mà không biết *nhãn* Y tương ứng.

Những thuật toán loại này được gọi là Unsupervised learning vì không giống như Supervised learning, chúng ta không biết câu trả lời chính xác cho mỗi dữ liệu đầu vào. Giống như khi ta học, không có thầy cô giáo nào chỉ cho ta biết đó là chữ A hay chữ B. Cụm *không giám sát* được đặt tên theo nghĩa này.

Phân nhóm các thuật toán của Machine Learning

Các bài toán Unsupervised learning được tiếp tục chia nhỏ thành hai loại:

1. Clustering (phân nhóm)
   Một bài toán phân nhóm toàn bộ dữ liệu X thành các nhóm nhỏ dựa trên sự liên quan giữa các dữ liệu trong mỗi nhóm. Ví dụ: phân nhóm khách hàng dựa trên hành vi mua hàng. Điều này cũng giống như việc ta đưa cho một đứa trẻ rất nhiều mảnh ghép với các hình thù và màu sắc khác nhau, ví dụ tam giác, vuông, tròn với màu xanh và đỏ, sau đó yêu cầu trẻ phân chúng thành từng nhóm. Mặc dù không cho trẻ biết mảnh nào tương ứng với hình nào hoặc màu nào, nhiều khả năng chúng vẫn có thể phân loại các mảnh ghép theo màu hoặc hình dạng.
2. Association

Phân nhóm các thuật toán của Machine Learning

Các bài toán Unsupervised learning được tiếp tục chia nhỏ thành hai loại:

1. Clustering (phân nhóm)
2. Association
   Là bài toán khi chúng ta muốn khám phá ra một quy luật dựa trên nhiều dữ liệu cho trước. Ví dụ: những khách hàng nam mua quần áo thường có xu hướng mua thêm đồng hồ hoặc thắt lưng; những khán giả xem phim Spider Man thường có xu hướng xem thêm phim Bat Man, dựa vào đó tạo ra một hệ thống gợi ý khách hàng (Recommendation System), thúc đẩy nhu cầu mua sắm.

Phân nhóm các thuật toán của Machine Learning

Semi-Supervised Learning (Học bán giám sát)

Các bài toán khi chúng ta có một lượng lớn dữ liệu X nhưng chỉ một phần trong chúng được gán nhãn được gọi là Semi-Supervised Learning. Những bài toán thuộc nhóm này nằm giữa hai nhóm được nêu bên trên.

Ví dụ

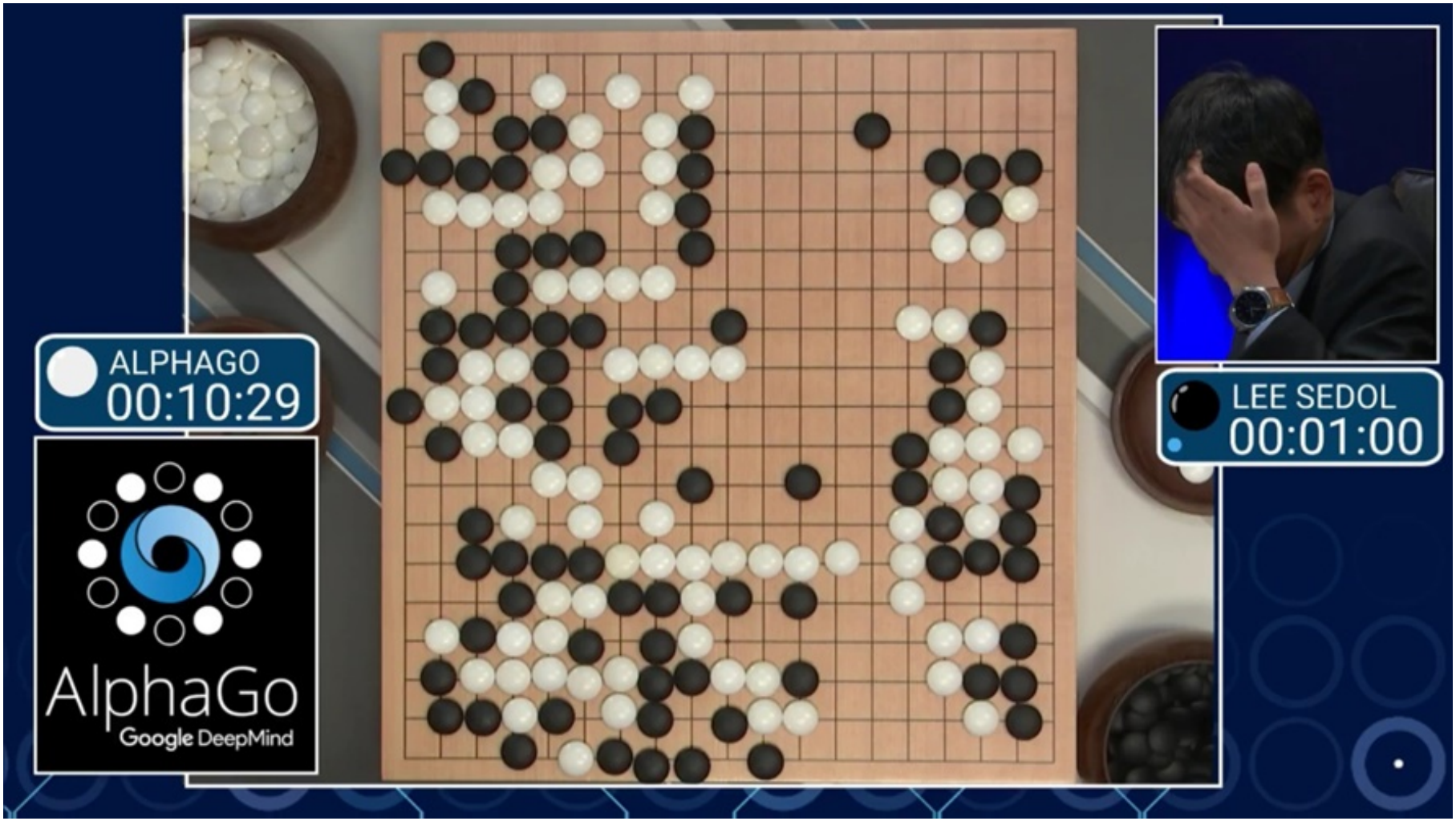
Một ví dụ điển hình của nhóm này là chỉ có một phần ảnh hoặc văn bản được gán nhãn (ví dụ bức ảnh về người, động vật hoặc các văn bản khoa học, chính trị) và phần lớn các bức ảnh/văn bản khác chưa được gán nhãn được thu thập từ internet. Thực tế cho thấy rất nhiều các bài toán Machine Learning thuộc vào nhóm này vì việc thu thập dữ liệu có nhãn tốn rất nhiều thời gian và có chi phí cao. Rất nhiều loại dữ liệu thậm chí cần phải có chuyên gia mới gán nhãn được (ảnh y học chẳng hạn). Ngược lại, dữ liệu chưa có nhãn có thể được thu thập với chi phí thấp từ internet.

Phân nhóm các thuật toán của Machine Learning

Reinforcement Learning (Học Củng Cố)

Reinforcement learning là các bài toán giúp cho một hệ thống tự động xác định hành vi dựa trên hoàn cảnh để đạt được lợi ích cao nhất (maximizing the performance). Hiện tại, Reinforcement learning chủ yếu được áp dụng vào Lý Thuyết Trò Chơi (Game Theory), các thuật toán cần xác định nước đi tiếp theo để đạt được điểm số cao nhất.

Phân nhóm các thuật toán của Machine Learning



AlphaGo chơi cờ vây với Lee Sedol. AlphaGo là một ví dụ của Reinforcement learning.

(Nguồn: AlphaGo AI Defeats Sedol Again, With "Near Perfect Game")

Phân nhóm các thuật toán của Machine Learning

Ví dụ (1)

AlphaGo gần đây nổi tiếng với việc chơi cờ vây thắng cả con người. Cờ vây được xem là có độ phức tạp cực kỳ cao với tổng số nước đi là xấp xỉ 10^{761}, *so với cờ vua là* 10^{120} *và tổng số nguyên tử trong toàn vũ trụ là khoảng* 10^{80}*!! Vì vậy, thuật toán phải chọn ra 1 nước đi tối ưu trong số hàng nhiều tỉ tỉ lựa chọn, và tất nhiên, không thể áp dụng thuật toán tương tự như IBM Deep Blue (IBM Deep Blue đã thắng con người trong môn cờ vua 20 năm trước). Về cơ bản, AlphaGo bao gồm các thuật toán thuộc cả Supervised learning và Reinforcement learning. Trong phần Supervised learning, dữ liệu từ các ván cờ do con người chơi với nhau được đưa vào để huấn luyện. Tuy nhiên, mục đích cuối cùng của AlphaGo không phải là chơi như con người mà phải thậm chí thắng cả con người. Vì vậy, sau khi học xong các ván cờ của con người, AlphaGo tự chơi với chính nó với hàng triệu ván chơi để tìm ra các nước đi mới tối ưu hơn. Thuật toán trong phần tự chơi này được xếp vào loại Reinforcement learning. (Xem thêm tại Google DeepMind's AlphaGo: How it works).*

Phân nhóm các thuật toán của Machine Learning

Ví dụ (2)

Huấn luyện cho máy tính chơi game Mario. Đây là một chương trình thú vị dạy máy tính chơi game Mario. Game này đơn giản hơn cờ vây vì tại một thời điểm, người chơi chỉ phải bấm một số lượng nhỏ các nút (di chuyển, nhảy, bắn đạn) hoặc không cần bấm nút nào. Đồng thời, phản ứng của máy cũng đơn giản hơn và lặp lại ở mỗi lần chơi (tại thời điểm cụ thể sẽ xuất hiện một chướng ngại vật cố định ở một vị trí cố định). Đầu vào của thuật toán là sơ đồ của màn hình tại thời điểm hiện tại, nhiệm vụ của thuật toán là với đầu vào đó, tổ hợp phím nào nên được bấm. Việc huấn luyện này được dựa trên điểm số cho việc di chuyển được bao xa trong thời gian bao lâu trong game, càng xa và càng nhanh thì được điểm thưởng càng cao (điểm thưởng này không phải là điểm của trò chơi mà là điểm do chính người lập trình tạo ra). Thông qua huấn luyện, thuật toán sẽ tìm ra một cách tối ưu để tối đa số điểm trên, qua đó đạt được mục đích cuối cùng là cứu công chúa.
Huấn luyện cho máy tính chơi game Mario.

Phân nhóm các thuật toán của Machine Learning

Phân nhóm dựa trên chức năng

Regression Algorithms

1. Linear Regression
2. Logistic Regression
3. Stepwise Regression

Classification Algorithms

1. Linear Classifier
2. Support Vector Machine (SVM)
3. Kernel SVM
4. Sparse Representation-based classification (SRC)

Instance-based Algorithms

1. k-Nearest Neighbor (kNN)
2. Learning Vector Quantization (LVQ)

Phân nhóm các thuật toán của Machine Learning

Phân nhóm dựa trên chức năng

Regularization Algorithms

1. Ridge Regression
2. Least Absolute Shrinkage and Selection Operator (LASSO)
3. Least-Angle Regression (LARS)

Bayesian Algorithms

1. Naive Bayes
2. Gaussian Naive Bayes (SRC)

Clustering Algorithms

1. k-Means clustering
2. k-Medians
3. Expectation Maximization (EM)

Phân nhóm các thuật toán của Machine Learning

Phân nhóm dựa trên chức năng

Artificial Neural Network Algorithms

1. Perceptron
2. Softmax Regression
3. Multi-layer Perceptron
4. Back-Propagation

Dimensionality Reduction Algorithms

1. Principal Component Analysis (PCA)
2. Linear Discriminant Analysis (LDA)

Ensemble Algorithms

1. Boosting
2. AdaBoost
3. Random Forest

1. Naive Bayes Classifier

Bài toán

Xét bài toán classification với C classes $1, 2, ..., C$. Giả sử có một điểm dữ liệu $x \in \mathbb{R}^d$. Hãy tính xác suất để điểm dữ liệu này rơi vào class c. Nói cách khác, hãy tính:

$$p(y = c|x) \tag{1}$$

hoặc viết gọn thành $p(c|x)$.

Tức tính xác suất để đầu ra là class c biết rằng đầu vào là vector x.

Biểu thức này, nếu tính được, sẽ giúp chúng ta xác định được xác suất để điểm dữ liệu rơi vào mỗi class. Từ đó có thể giúp xác định class của điểm dữ liệu đó bằng cách chọn ra class có xác suất cao nhất:

$$c = \arg \max_{c \in \{1,...,C\}} p(c|x) \tag{2}$$

Naive Bayes Classifier

Biểu thức (2) thường khó được tính trực tiếp. Thay vào đó, quy tắc Bayes thường được sử dụng:

$$c = \arg\max_c p(c|x) \tag{3}$$

$$= \arg\max_c \frac{p(x|c)p(c)}{p(x)} \tag{4}$$

$$= \arg\max_c p(x|c)p(c) \tag{5}$$

Từ (3) sang (4) là vì quy tắc Bayes. Từ (4) sang (5) là vì mẫu số $p(x)$ không phụ thuộc vào c.

Tiếp tục xét biểu thức (5), $p(c)$ có thể được hiểu là xác suất để một điểm rơi vào class c. Giá trị này có thể được tính bằng MLE, tức tỉ lệ số điểm dữ liệu trong tập training rơi vào class này chia cho tổng số lượng dữ liệu trong tập traing; hoặc cũng có thể được đánh giá bằng MAP estimation. Trường hợp thứ nhất thường được sử dụng nhiều hơn.

Naive Bayes Classifier

Thành phần còn lại $p(x|c)$, tức phân phối của các điểm dữ liệu trong class c, thường rất khó tính toán vì x là một biến ngẫu nhiên nhiều chiều, cần rất rất nhiều dữ liệu training để có thể xây dựng được phân phối đó. Để giúp cho việc tính toán được đơn giản, người ta thường giả sử một cách đơn giản nhất rằng các thành phần của biến ngẫu nhiên x là độc lập với nhau, nếu biết c (given c). Tức là:

$$p(x|c) = p(x_1, x_2, ..., x_d|c) = \prod_{i=1}^{p} p(x_i|c) \tag{6}$$

Giả thiết các chiều của dữ liệu độc lập với nhau, nếu biết cc, là quá chặt và ít khi tìm được dữ liệu mà các thành phần hoàn toàn độc lập với nhau. Tuy nhiên, giả thiết *ngây ngô* này lại mang lại những kết quả tốt bất ngờ. Giả thiết về sự độc lập của các chiều dữ liệu này được gọi là *Naive Bayes* (xin không dịch). Cách xác định class của dữ liệu dựa trên giả thiết này có tên là *Naive Bayes Classifier* (NBC).

Naive Bayes Classifier

NBC, nhờ vào tính đơn giản một cách *ngây thơ*, có tốc độ training và test rất nhanh. Việc này giúp nó mang lại hiệu quả cao trong các bài toán large-scale.

- Ở bước **training**, các phân phối $p(c)$ và $p(x_i|c)$, $i = 1, ..., d$ sẽ được xác định dựa vào training data. Việc xác định các giá trị này có thể dựa vào Maximum Likelihood Estimation hoặc Maximum A Posteriori.
- Ở bước **test**, với một điểm dữ liệu mới x, class của nó sẽ được xác đinh bởi:

$$c = \arg \max_{c \in \{1,...,C\}} p(c) \prod_{i=1}^{p} p(x_i|c) \tag{7}$$

Naive Bayes Classifier

Khi d lớn và các xác suất nhỏ, biểu thức ở vế phải của (7) sẽ là một số rất nhỏ, khi tính toán có thể gặp sai số. Để giải quyết việc này, (7) thường được viết lại dưới dạng tương đương bằng cách lấy log của vế phải:

$$c = \arg \max_{c \in \{1,\dots,C\}} \log(p(c)) + \sum_{i=1}^{d} \log(p(x_i|c)) \tag{7.1}$$

Việc này không ảnh hưởng tới kết quả vì log là một hàm đồng biến trên tập các số dương.

Mặc dù giả thiết mà Naive Bayes Classifiers sử dụng là quá phi thực tế, chúng vẫn hoạt động khá hiệu quả trong nhiều bài toán thực tế, đặc biệt là trong các bài toán phân loại văn bản, ví dụ như lọc tin nhắn rác hay lọc email spam. Trong phần sau của bài viết, chúng ta cùng xây dựng một bộ lọc email spam tiếng Anh đơn giản.

Naive Bayes Classifier

Cả việc training và test của NBC là cực kỳ nhanh khi so với các phương pháp classification phức tạp khác. Việc giả sử các thành phần trong dữ liệu là độc lập với nhau, nếu biết class, khiến cho việc tính toán mỗi phân phối $p(x_i|c)$ trở nên cực kỳ nhanh.

Mỗi giá trị $p(c)$, $c = 1, 2, ..., C$ có thể được xác định như là tần suất xuất hiện của class c trong training data.

Việc tính toán $p(x_i|c)$ phụ thuộc vào loại dữ liệu. Có ba loại được sử dụng phổ biến là: Gaussian Naive Bayes, Multinomial Naive Bayes, và Bernoulli Naive.

2. Các phân phối thường dùng cho $p(x_i|c)$

Mục này chủ yếu được dịch từ tài liệu của thư viện sklearn.

Gaussian Naive Bayes

Mô hình này được sử dụng chủ yếu trong loại dữ liệu mà các thành phần là các biến liên tục.

Với mỗi chiều dữ liệu i và một class c, x_i tuân theo một phân phối chuẩn có kỳ vọng μ_{ci} và phương sai σ_{ci}^2:

$$p(x_i|c) = p(x_i|\mu_{ci}, \sigma_{ci}^2) = \frac{1}{\sqrt{2\pi\sigma_{ci}^2}} \exp\left(-\frac{(x_i - \mu_{ci})^2}{2\sigma_{ci}^2}\right) \tag{8}$$

Trong đó, bộ tham số $\theta = \{\mu_{ci}, \sigma_{ci}^2\}$ được xác định bằng Maximum Likelihood:

$$(\mu_{ci}, \sigma_{ci}^2) = \arg\max_{\mu_{ci}, \sigma_{ci}^2} \prod_{n=1}^{N} p(x_i^{(n)}|\mu_{ci}, \sigma_{ci}^2) \tag{9}$$

Naive Bayes Classifier

Gaussian Naive Bayes

Đây là cách tính của thư viện sklearn. Chúng ta cũng có thể đánh giá các tham số bằng MAP nếu biết trước priors của μ_{ci} *và* σ^2_{ci}

Multinomial Naive Bayes

Mô hình này chủ yếu được sử dụng trong phân loại văn bản mà feature vectors được tính bằng Bags of Words. Lúc này, mỗi văn bản được biểu diễn bởi một vector có độ dài d chính là số từ trong từ điển. Giá trị của thành phần thứ i trong mỗi vector chính là số lần từ thứ i xuất hiện trong văn bản đó.

Khi đó, $p(x_i|c)$ tỉ lệ với tần suất từ thứ i (hay feature thứ i cho trường hợp tổng quát) xuất hiện trong các văn bản của class c. Giá trị này có thể được tính bằng cách:

$$\lambda_{ci} = p(x_i|c) = \frac{N_{ci}}{N_c} \tag{10}$$

Multinomial Naive Bayes

Trong đó:

- N_{ci} là tổng số lần từ thứ i xuất hiện trong các văn bản của class c, nó được tính là tổng của tất cả các thành phần thứ i của các feature vectors ứng với class c.
- N_c là tổng số từ (kể cả lặp) xuất hiện trong class c. Nói cách khác, nó bằng tổng độ dài của toàn bộ các văn bản thuộc vào class c. Có thể suy ra rằng $N_c = \sum_{i=1}^{d} N_{ci}$, từ đó $\sum_{i=1}^{d} \lambda_{ci} = 1$.

Cách tính này có một hạn chế là nếu có một từ mới chưa bao giờ xuất hiện trong class c thì biểu thức (10) sẽ bằng 0, điều này dẫn đến vế phải của (7) bằng 0 bất kể các giá trị còn lại có lớn thế nào. Việc này sẽ dẫn đến kết quả không chính xác (xem thêm ví dụ ở mục sau).

Multinomial Naive Bayes

Để giải quyết việc này, một kỹ thuật được gọi là *Laplace smoothing* được áp dụng:

$$\hat{\lambda}_{ci} = \frac{N_{ci} + \alpha}{Nc + d\alpha} \tag{11}$$

Với α là một số dương, thường bằng 1, để tránh trường hợp tử số bằng 0. Mẫu số được cộng với $d\alpha$ để đảm bảo tổng xác suất $\sum_{i=1}^{d} \hat{\lambda}_{ci} = 1$.
Như vậy, mỗi class c sẽ được mô tả bởi bộ các số dương có tổng bằng 1:

$$\hat{\lambda}_c = \{\hat{\lambda}_{c1}, ..., \hat{\lambda}_{cd}\}.$$

Bernoulli Naive Bayes

Mô hình này được áp dụng cho các loại dữ liệu mà mỗi thành phần là một giá trị binary - bằng 0 hoặc 1. Ví dụ: cũng với loại văn bản nhưng thay vì đếm tổng số lần xuất hiện của 1 từ trong văn bản, ta chỉ cần quan tâm từ đó có xuất hiện hay không.

Khi đó, $p(x_i|c)$ được tính bằng:

$$p(x_i|c) = p(i|c)^{x_i}(1 - p(i|c))^{1-x_i}$$

với $p(i|c)$ có thể được hiểu là xác suất từ thứ i xuất hiện trong các văn bản của class c.

Tài liệu tham khảo

1. https://vi.wikipedia.org/wiki
2. https://machinelearningcoban.com
3. GS. Nguyễn Hùng Sơn, Ba Lan, Neural Turing Machine, Khóa học về khoa học dữ liệu, Tháng 01/2019, Đà Nẵng.
4. GS. Phùng Quốc Định, Australia, Generative Deep Models, Khóa học về khoa học dữ liệu, Tháng 01/2019, Đà Nẵng.
5. GS. Hồ Tú Bảo, Trí tuệ nhân tạo và khoa học Dữ liệu trong thời chuyển đổi số, Khóa học về khoa học dữ liệu, Tháng 01/2019, Đà Nẵng.