# Học máy là gì ? Tại sao và khi nào phải dùng máy học ?

Lập trình truyền thống và lập trình máy học.

Khi lập trình truyền thống không thể hoặc gặp khó khan khi giải quyết các bài toán như dự đoán một kết quả có thể xảy ra dựa vào tập các kết quả tương tự có trước, khi đó, chúng ta cần dùng đến học máy.

Vậy khi nào thì dùng được học máy ?

Phải tồn tại dạng mẫu

Không thể đưa ra công thức hay quy luật rõ rang cho vấn đề

Phải có tập dữ liệu

Học máy là xây dựng một chương trình máy tính có khả năng tự cải thiện hiệu năng bằng kinh nghiệm.

Một trong những đặc trưng thú vị của học máy là nó là sự giao thoa của các ngành ngành: khoa học máy tính, thống kê học, toán học, và kĩ thuật.

Giá trị quan trọng của học máy là mô hình dự đoán, ở đó chúng ta dùng các dữ liệu quá khứ để huấn luyện mô hình và dùng mô hình để dự đoán. Nói một cách ngắn gọn, đó là viết một chương trình có thể tự học.

Học máy có nhiều tên gọi khác nhau, được đặt bởi những cộng đồng nghiên cứu khác nhau. Khi sử dụng học máy để nhận dạng các mẫu, học máy là pattern recognition; khi sử dụng trong khai phá dữ liệu, học máy là data mining; khi sử dụng trong các mô hình dự đoán, học máy có tên là redictive analytics hay còn gọi là predictive modeling; khi sử dụng vào phân tích dữ liệu, học máy có tên gọi là data science; khi sử dụng trong các hệ thống tự thích nghi tương tác với môi trường, học máy được gọi là adaptive systems; khi sử dụng để khai phá tri thức, học máy có tên gọi là knowledge discovery; khi sử dụng trong việc phân tích và thống kê dữ liệu, học máy có tên gọi là statistic learning hay còn gọi là statistic modeling. Ngoài ra còn có thể kể ra rất nhiều tên khác nưa: statistical modeling, self-organizing systems, v.v…

Đứng ở góc độ người mới tìm hiểu và muốn nghiên cứu hay xây dựng cho mình một chương trình học máy, chúng ta cần:

1. Sử dụng được một ngôn ngữ lập trình. Ví dụ như Java, Matlab, R, Python, ... Tuy nhiên, Python là một lựa chọn rất phù hợp

2. Bổ sung một vài kiến thức về toán học cho khoa học máy tính như: Lý thuyết xác suất thống kê, đại số tuyến tính.

3. Tìm hiểu về lĩnh vực chuyên sâu mà bạn muốn xây dựng ứng dụng, ví dụ: y học (dự đoán bệnh), nông nghiệp (dự báo năng suất), ...

# Phân loại

## Phân loại dựa trên phương thức học

### Supervised Learning(Học có giám sát)

Supervised learning là một hướng tiếp cận của machine learning để làm cho máy tính có khả năng "học". Trong hướng tiếp cận này, người ta "huấn luyện" máy tính dựa trên những quan sát có dán nhãn(labeled data)

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.

Trong supervised learning, các quan sát bắt buộc phải được dán nhãn trước

VD: Nhận diện chữ viết tay, nhận diện khuân mặt.

#### 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.

#### Regression (Hồi quy)

Nếu label không được chia thành các nhóm mà là một giá trị thực cụ thể. Ví dụ: một căn nhà rộng x m2, có y phòng ngủ và cách trung tâm thành phố z km có giá bn ?

Phần dự đoán giới tính có thể coi là thuật toán Classification, phần dự đoán tuổi có thể coi là thuật toán Regression.

### 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.

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.

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

#### 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.

#### 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.

### 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.

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.

### 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 loại dựa trên chức năng

Phân nhóm dựa trên chức năng

Có một cách phân nhóm thứ hai dựa trên chức năng của các thuật toán. Trong phần này, tôi xin chỉ liệt kê các thuật toán. Thông tin cụ thể sẽ được trình bày trong các bài viết khác tại blog này. Trong quá trình viết, tôi có thể sẽ thêm bớt một số thuật toán.

### Regression Algorithms

#### Linear Regression(Hồi quy tuyến tính)

<https://machinelearningcoban.com/2016/12/28/linearregression/>

Có thể áp dụng với các dữ liệu training theo mô hình tuyến tính

Nhạy cảm với nhiễu=> cần loại bỏ nhiễu(tiền xử lí)

Không biểu diễn được các mô hình phức tạp.

#### Logistic Regression

<https://machinelearningcoban.com/2017/01/04/kmeans2/>

#### Stepwise Regression

### Classification Algorithms

#### Linear Classifier

#### Support Vector Machine (SVM)

#### Kernel SVM

#### Sparse Representation-based classification (SRC)

### Instance-based Algorithms

#### k-Nearest Neighbor (kNN)

<https://machinelearningcoban.com/2017/01/08/knn/>

K-nearest neighbor là một trong những thuật toán supervised-learning đơn giản nhất (mà hiệu quả trong một vài trường hợp) trong Machine Learning. Khi training, thuật toán này không học một điều gì từ dữ liệu training (đây cũng là lý do thuật toán này được xếp vào loại lazy learning), mọi tính toán được thực hiện khi nó cần dự đoán kết quả của dữ liệu mới. K-nearest neighbor có thể áp dụng được vào cả hai loại của bài toán Supervised learning là Classification và Regression. KNN còn được gọi là một thuật toán Instance-based hay Memory-based learning.

Với KNN, trong bài toán Classification, label của một điểm dữ liệu mới (hay kết quả của câu hỏi trong bài thi) được suy ra trực tiếp từ K điểm dữ liệu gần nhất trong training set. Label của một test data có thể được quyết định bằng major voting (bầu chọn theo số phiếu) giữa các điểm gần nhất, hoặc nó có thể được suy ra bằng cách đánh trọng số khác nhau cho mỗi trong các điểm gần nhất đó rồi suy ra label

Trong bài toán Regresssion, đầu ra của một điểm dữ liệu sẽ bằng chính đầu ra của điểm dữ liệu đã biết gần nhất (trong trường hợp K=1), hoặc là trung bình có trọng số của đầu ra của những điểm gần nhất, hoặc bằng một mối quan hệ dựa trên khoảng cách tới các điểm gần nhất đó.

Một cách ngắn gọn, KNN là thuật toán đi tìm đầu ra của một điểm dữ liệu mới bằng cách chỉ dựa trên thông tin của K điểm dữ liệu trong training set gần nó nhất (K-lân cận), không quan tâm đến việc có một vài điểm dữ liệu trong những điểm gần nhất này là nhiễu.

Có một điều đáng lưu ý là KNN phải nhớ tất cả các điểm dữ liệu training, việc này không được lợi về cả bộ nhớ và thời gian tính toán

KNN rất nhạy cảm với nhiễu khi K nhỏ.

Như đã nói, KNN là một thuật toán mà mọi tính toán đều nằm ở khâu test. Trong đó việc tính khoảng cách tới từng điểm dữ liệu trong training set sẽ tốn rất nhiều thời gian, đặc biệt là với các cơ sở dữ liệu có số chiều lớn và có nhiều điểm dữ liệu. Với K càng lớn thì độ phức tạp cũng sẽ tăng lên. Ngoài ra, việc lưu toàn bộ dữ liệu trong bộ nhớ cũng ảnh hưởng tới hiệu năng của KNN

#### Learning Vector Quantization (LVQ)

### Regularization Algorithms

#### Ridge Regression

#### Least Absolute Shrinkage and Selection Operator (LASSO)

#### Least-Angle Regression (LARS)

### Bayesian Algorithms

#### Naive Bayes

#### Gaussian Naive Bayes

### Clustering Algorithms

#### k-Means clustering(phân cụm K-means)

<https://machinelearningcoban.com/2017/01/01/kmeans/>

Mục đích cuối cùng của thuật toán phân nhóm này là: từ dữ liệu đầu vào và số lượng nhóm chúng ta muốn tìm, hãy chỉ ra center của mỗi nhóm và phân các điểm dữ liệu vào các nhóm tương ứng. Giả sử thêm rằng mỗi điểm dữ liệu chỉ thuộc vào đúng một nhóm.

Thử nghiệm <https://www.naftaliharris.com/blog/visualizing-k-means-clustering/>

#### k-Medians

#### Expectation Maximization (EM)

### Artificial Neural Network Algorithms

#### Perceptron

<https://machinelearningcoban.com/2017/01/21/perceptron/>

Perceptron là một thuật toán Classification cho trường hợp đơn giản nhất: chỉ có hai class (lớp) (bài toán với chỉ hai class được gọi là binary classification) và cũng chỉ hoạt động được trong một trường hợp rất cụ thể. Tuy nhiên, nó là nền tảng cho một mảng lớn quan trọng của Machine Learning là Neural Networks và sau này là Deep Learning.

Tóm tắt PLA

1. Chọn ngẫu nhiên một vector hệ số w với các phần tử gần 0.
2. Duyệt ngẫu nhiên qua từng điểm dữ liệu xi:
   * Nếu xi được phân lớp đúng, tức sgn(wTxi)=yisgn(wTxi)=yi, chúng ta không cần làm gì.
   * Nếu xixi bị misclassifed, cập nhật ww theo công thức:w=w+yixiw=w+yixi
3. Kiểm tra xem có bao nhiêu điểm bị misclassifed. Nếu không còn điểm nào, dừng thuật toán. Nếu còn, quay lại bước 2.

#### Softmax Regression

#### Multi-layer Perceptron

#### Back-Propagation

### Dimensionality Reduction Algorithms

#### Principal Component Analysis (PCA)

#### Linear Discriminant Analysis (LDA)

### Ensemble Algorithms

#### Boosting

#### AdaBoost

#### Random Forest