ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

NGUYỄN VIỆT THÀNH - 15520814
LÊ HOÀNG TUẤN - 15520967

ĐỒ ÁN CHUYÊN NGÀNH

ÁP DỤNG MÁY HỌC ĐỂ PHÁT HIỆN TRAFFIC BẤT THƯỜNG

GIẢNG VIÊN HƯỚNG DẪN
ThS. UNG VĂN GIÀU

TP. HỒ CHÍ MINH, 2018

Mục lục

Danh sách hình vẽ

Danh sách bảng

TÓM TẮT ĐỒ ÁN

Trong bài báo cáo này chúng tôi trình bày về vấn đề ứng dụng công nghệ máy học (hay còn gọi là học máy - machine learning) vào việc phân tích và phát hiện các luồng traffic nào là bình thường và bất thường. Trong phạm vi của đồ án, chúng tôi tập trung vào phân tích các HTTP request từ tập dữ liệu CSIC 2010. Kết quả đạt được của chúng tôi là phân tích các traffic nào là bất thường và bình thường, thuật toán mà chúng tôi chọn là Decision Tree.

Nội dung của bài báo cáo này gồm 2 phần chính là:

- **Cơ sở lý thuyết** - phần này chúng tôi sẽ giới thiệu một số lý thuyết toán học liên quan. Sau đó chúng tôi giới thiệu sơ lược về máy học, các khái niệm, và phân loại. Tiếp theo chúng tôi cũng phân tích thuật toán mà chúng tôi chọn sử dụng - Decision Tree. Bên cạnh đó, chúng tôi cũng tiếp cận một số lỗ hổng bảo mật web phổ biến.
- **Áp dụng thuật toán vào phân tích tập dữ liệu** - phần này là kết quả của nhóm chúng tôi đạt được, phần nãy sẽ tập trung vào tập dữ liệu và thuật toán mà chúng tôi chọn sử dụng.

Trong bài báo cáo, chúng tôi sử dụng một số thuật ngữ tiếng Anh thay vì dịch ra tiếng Việt.

MỞ ĐẦU

Lý do chọn đề tài

Nhiều ứng dụng web ngày nay gặp vấn đề bảo mật, nguyên nhân nó từ các nhà phát triển ứng dụng web, muốn tạo ra sản phẩm nhanh, không quan tâm cũng như kiến thức liên quan đến bảo mật. Để khắc phục vấn đề bảo mật. Nhà phát triển web cần tìm ra một công cụ để giảm thiểu rủi ro bảo mật. Phát hiện xâm nhập là một công cụ mạnh mẽ để nhận diện và ngăn chặn tấn công tới hệ thống. Hầu hết những công nghệ phát hiện xâm nhập hệ thống web hiện nay không có khả năng giải quyết các tấn công web phức tạp, những kiểu tấn công mới chưa từng biết trước đó.

Tuy nhiên, với việc áp dụng máy học (tiếng anh: **machine learning**), ta có thể xây dựng những mô hình giúp phát hiện những kiểu tấn công đã biết hoặc chưa biết. Như chúng ta đã biết, 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.

Chính vì những điều trên đã thôi thúc chúng tôi tiến hành tiếp cận máy học trong lĩnh vực phát hiện tấn công web.

Mục đích thực hiện đề tài

Khi thực hiện đề tài, nhóm chúng tôi mong muốn được tiếp cận nghiên cứu và tìm hiểu về lĩnh vực máy học. Và từ đó vận dụng vào ngành mà chúng tôi đang học - An toàn thông tin.

Hai mục tiêu mà chúng tôi hướng đến để đạt được trong đề tài này là:

- Thứ nhất, sẽ có kiến thức cơ bản về máy học và lý thuyết liên quan.
- Thứ hai, tìm hiểu và chọn được một thuật toán để vận dụng phân tích một tập dữ liệu cho trước để phát hiện luồng traffic nào là bình thường và bất thường.

Đối tượng và phạm vi nghiên cứu của đề tài

Đối tượng và phạm vi nghiên cứu của chúng tôi tập trung vào hai điểm chính:

- **Tập dữ liệu được sử dụng để training** - ở đây chúng tôi chọn tập dữ liệu **HTTP DATASET CSIC 2010**. Lý do vì sao chúng tôi chọn tập dữ liệu này sẽ được trình bày chi tiết trong phần sau của báo cáo.
- **Thuật toán được sử dụng** - thuật toán mà nhóm chúng tôi chọn là Decision Tree. Lý do nhóm chọn cũng sẽ được giới thiệu chi tiết trong phần nội dung của bài báo cáo

Trong phạm vi của đồ án, nhóm chúng tôi chỉ tập trung vào việc phân tích GET và POST trong thành phần HTTP Header của các gói tin.

Chương 1

CƠ SỞ LÝ THUYẾT

1.1 Một số lý thuyết về xác suất

Có thể nói một điều rằng lý thuyết xác suất là một trong những lý thuyết quan trọng nhất của khoa học hiện đại và đặc biệt là **machine learning** bởi vì đa phần các thuật toán của Machine Learning đều có cơ sở dựa trên xác suất.

Phần bên dưới chúng tôi trình bày chủ yếu một số lý thuyết cơ bản được chúng tôi tìm hiểu và tổng hợp từ [2]

1.1.1 Không gian xác suất

Khi nói đến xác suất là người ta nói đến các lý thuyết toán học về sự *bất định - uncertainly* hay nói một cách khác, xác suất biểu thị khả năng xảy ra của các *sự kiện - event* trong một môi trường bất định nào đó. Ví dụ chúng ta xét về xác suất có mưa hay không có mưa vào thứ hai tuần tới, xác suất tỏ tình thành công hay thất bại của cậu bạn thân,... Tóm lại cứ nói đến xác suất là đề cập đến sự không chắc chắn hay bất định đó.

Về mặt toán học, người ta kí hiệu một **không gian xác suất - probability space** bao gồm 3 thành phần (Ω, F, P) như sau:

- Ω (có thể đọc là "Ô-me-ga") chính là tập các giá trị **có thể xảy ra - possible outcome**

với sự kiện trong không gian xác suất. Người ta còn gọi nó là **không gian mẫu**.

- $F \subseteq 2^{\Omega}$ là tập hợp các sự kiện có thể xảy ra trong không gian xác suất.
- P là xác suất (hoặc phân phối xác suất) của sự kiện. P ánh xạ một sự kiện $E \in F$ vào trong một giá trị thực $p \in [0; 1]$. Ở đây chúng ta gọi $p = P(E)$ là xác suất của sự kiện E.

Chúng ta cùng nhau xem xét một ví dụ khá kinh điển trong lý thuyết xác suất đó chính là ví dụ **tung xúc sắc**.

Ví dụ 1.1. Giả sử rằng chúng ta tung một con xúc sắc 6 mặt. Không gian các **outcomes** có thể xảy ra trong trường hợp này là $\Omega = \{1, 2, 3, 4, 5, 6\}$ - chúng ta không tính đến các trường hợp xúc sắc rơi lơ lửng tức là không thuộc mặt nào. Không gian các sự kiện F sẽ tùy thuộc vào sự định nghĩa của chúng ta. Ví dụ chúng ta định nghĩa sự kiện xúc sắc là mặt chẵn hoặc mặt lẻ thì không gian sự kiện $F = \{\varnothing, \{1, 3, 5\}, \{2, 4, 6\}, \Omega\}$ trong đó $\varnothing$ là sự kiện có xác suất 0 - hay còn gọi là biến cố *không thể có*. Ω là sự kiện có xác suất 1 - hay còn gọi là *biến cố chắc chắn*.

1.1.2 Các tính chất xác suất

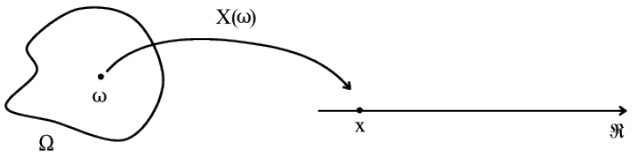
Giống như ví dụ ở phía trên, khi *không gian mẫu - outcomes space* là hữu hạn thì chúng ta thường lựa chọn không gian sự kiện $F = 2^{\Omega} = \{\varnothing, \{1, 3, 5\}, \{2, 4, 6\}, \Omega\}$. Cách tiếp cận này chưa hẳn đã tổng quát hóa cho mọi trường hợp tuy nhiên nó đủ dùng trong các bài toán thực tế, tất nhiên là với giả thiết không gian mẫu của chúng ta là **hữu hạn**. Khi không gian mẫu là **vô hạn - infinite** chúng ta phải hết sức cẩn thận trong việc lựa chọn không gian sự kiện F. Khi đã định nghĩa được không gian sự kiện F thì hàm xác suất của chúng ta bắt buộc phải thỏa mãn các tính chất sau đây:

- **Không âm - non-negativity** - xác suất của mọi sự kiện là không âm, tức là với mọi $x \in F, \;\; P(x) \geq 0$
- **Xác suất toàn cục - trivial event** $P(\Omega) = 1$

- **Tính cộng - additivity** tức là với mọi $x, y \in F$ nếu như $x \cap y = \varnothing$ thì ta có $P(x \cup y) = P(x) + P(y)$

1.1.3 Biến ngẫu nghiên

Biến ngẫu nghiên (Random Variables) là một thành phần quan trọng trong lý thuyết xác suất. Nó biểu diễn giá trị của các đại lượng không xác định, thông thường nó được coi như một ánh xạ từ tập các **outcomes** trong không gian mẫu thành các giá trị thực.



Hình 1.1: Minh họa giá trị X ánh xạ từ các tập outcomes

Quay trở lại với ví dụ tung xúc sắc phía trên, gọi X là biến ngẫu nhiên biểu diễn kết quả của các những lần gieo xúc sắc. Một lựa chọn khá tự nhiên và đơn giản đó là: **"X là số chấm tròn trên mặt tung được"**

Chúng ta cũng có thể lựa chọn một chiến lược biểu diễn biến ngẫu nhiên X khác chẳng hạn như sau:

$$X = \begin{cases} 1 & \text{if i is odd} \\ 0 & \text{if i is even} \end{cases} \tag{1.1}$$

Có nghĩa là cùng một biến cố nhưng biểu diễn nó như thế nào là việc của mỗi chúng ta. Biến ngẫu nhiên X biểu diễn như biểu thức (1.1) được gọi là *binary random variables - biến nhị phân*. Biến nhị phân được sử dụng rất thông dụng trong thực tế công việc nhất là Machine Learning và thường được biết đến với cái tên **indicator variables** nó thể hiện s*ự xảy ra hay không xảy ra* của một sự kiện.

Biến ngẫu nhiên rời rạc và biến ngẫu nhiên liên tục

Có hai loại biến ngẫu nhiên đó là **BNN rời rạc** (discrete) và **BNN liên tục** (continuous).

Rời rạc có thể hiểu một cách đơn giản là giá trị của nó thuộc vào một tập định trước. Ví dụ tung đồng xu thì có hai khả năng là head và tail [1]. Tập các giá trị này có thể là có thứ tự (khi tung xúc xắc) hoặc không có thứ tự (unorderd), ví dụ khi đầu ra là các giá trị *nắng, mưa, bão*,.... Mỗi đầu ra có một giá trị xác suất tương ứng với nó. Trong Machine Learning các giá trị này tương ứng với *các phân lớp (class)*. Các giá trị xác suất này không âm và có tổng bằng một:

$$\sum_{\forall x} p(x) = 1 \tag{1.2}$$

Còn **biến ngẫu nhiên liên tục** có thể định nghĩa là các biến ngẫu nhiên mà các giá trị của nó rơi vào một tập *không biết trước*. Trong Machine Learning người ta gọi lớp bài toán với biến ngẫu nhiên liên tục là **Hồi quy**. Giá trị của nó có thể nằm trong một khoảng hữu hạn ví dụ như thời gian làm bài thi đại học là $t \in (0; 180)$ phút hoặc cũng có thể là vô hạn ví dụ như thời gian từ bây giờ đến ngày tận thế $t \in (0; +\infty)$ chẳng hạn. Khi đó hàm mật độ xác suất của nó trên toàn miền giá trị D của outcomes space được định nghĩa bằng một tích phân như sau:

$$\int_D p(x)dx = 1 \tag{1.3}$$

1.1.4 Xác suất có điều kiện

Dựa vào phổ điểm của các học sinh, liệu ta có thể tính được xác suất để một học sinh được điểm 10 môn Lý, biết rằng học sinh đó được điểm 1 môn Toán (ai cũng có quyền hy vọng). Hoặc biết rằng bây giờ đang là tháng 7, tính xác suất để nhiệt độ hôm nay cao hơn 30

[1] Tên gọi này bắt nguồn từ đồng xu Mỹ, một mặt có hình mặt người, được gọi là *head*, trái ngược với mặt này được gọi là mặt *tail*, cách gọi này hay hơn cách gọi *xấp ngửa* vì ta không có quy định rõ ràng thế nào là xấp ngay ngửa

độ C.

Xác suất có điều kiện (**conditional probability**) của một biến ngẫu nhiên x biết rằng biến ngẫu nhiên y có giá trị y^* được ký hiệu là $p(x|y = y^*)$ (đọc là "*xác suất của* x *biết* y *có giá trị* y^*" - *probability of* x *given that* y *takes value* y^*).

Xác suất có điều kiện $p(x|y = y^*)$ có thể được tính dựa trên *joint probobability* $p(x, y)$ [2]. Tổng quát ta có công thức để tính như sau:

$$p(x|y = y^*) = \frac{p(x, y = y^*)}{\sum_x p(x, y = y^*)} = \frac{p(x, y = y^*)}{p(y = y^*)} \tag{1.4}$$

Thông thường, ta có thể viết xác suất có điều kiện mà không cần chỉ rõ giá trị $y = y^*$ và có công thức gọn hơn:

$$p(x|y) = \frac{p(x, y)}{p(y)} \tag{1.5}$$

Tương tự:

$$p(y|x) = \frac{p(y, x)}{p(x)} \tag{1.6}$$

Và ta sẽ có quan hệ:

$$p(x, y) = p(x|y)p(y) = p(y|x)p(x) \tag{1.7}$$

Khi có nhiều hơn hai biến ngẫu nhiên, ta có các công thức:

$$\begin{aligned} p(x, y, z, w) &= p(x, y, z|w)p(w) && (1.8)\\ &= p(x, y|z, w)p(z, w) = p(x, y|z, w)p(z|w)p(w) && (1.9)\\ &= p(x|y, z, w)p(y|z, w)p(z|w)p(w) && (1.10) \end{aligned}$$

Công thức 1.10 có dạng chuỗi (*chain*) và được sử dụng nhiều sau này.

[2] Xác suất hợp (*Joint probability*) là xác suất của hai biến cố cùng xảy ra.

1.1.5 Quy tắc Bayes

Công thức (1.7) biểu diễn *joint probability* theo hai cách. Từ đây ta có thể suy ra quan hệ giữa hai *conditional probabilities* $p(x|y)$ và $p(y|x)$:

$$p(y|x)p(x) = p(x|y)p(y)$$

Biến đổi một chút:

$$\begin{aligned} p(y|x) &= \frac{p(x|y)p(y)}{p(x)} && (1.11)\\ &= \frac{p(x|y)p(y)}{\sum_y p(x,y)} && (1.12)\\ &= \frac{p(x|y)p(y)}{\sum_y p(x|y)p(y)} && (1.13) \end{aligned}$$

Từ (1.13) ta có thể thấy rằng $p(y|x)$ hoàn toàn có thể tính được nếu ta biết mọi $p(x|y)$ và $p(y)$. Tuy nhiên, việc tính trực tiếp xác suất này thường là phức tạp. Thay vào đó, ta có thể đi tìm mô hình phù hợp của $p(\mathbf{x}|y)$ trên training data sao cho *những gì đã thực sự xảy ra có xác suất cao nhất có thể*. Dựa trên training data, các tham số của mô hình này có thể tìm được qua một *bài toán tối ưu*.

Ba công thức (1.11) - (1.13) thường được gọi là Quy tắc Bayes (Bayes' rule). Quy tắc này rất quan trọng trong Machine Learning!

Trong Machine Learning, chúng ta thường mô tả quan hệ giữa hai biến x và y dưới dạng xác suất có điều kiện $p(x|y)$. Ví dụ, biết rằng đầu vào là một bức ảnh ở dạng vector $\vec{x}$, xác suất để bức ảnh chứa một chiếc xe là bao nhiêu. Khi đó, ta phải tính $p(y|\vec{x})$.

Độc lập (Independence)

Nếu biết giá trị của một biến ngẫu nhiên x không mang lại thông tin về việc suy ra giá trị của biến ngẫu nhiên y (và ngược lại), thì ta nói rằng hai biến ngẫu nhiên là *độc lập*

(independence). Chẳng hạn, chiều cao của một học sinh và điểm thi môn Toán của học sinh đó có thể coi là hai biến ngẫu nhiên độc lập.

Khi hai biến ngẫu nhiên x và y là *độc lập*, ta sẽ có:

$$p(x|y) = p(x) \tag{1.14}$$

$$p(y|x) = p(y) \tag{1.15}$$

Thay vào biểu thức Conditional Probability trong (1.7), ta có:

$$p(x, y) = p(x|y)p(y) = p(x)p(y) \tag{1.16}$$

1.1.6 Kỳ vọng

Kỳ vọng (expectation) của một biến ngẫu nhiên được định nghĩa là:

$$\mathrm{E}[x] = \sum_x xp(x) \qquad \text{if } x \text{ is discrete} \tag{1.17}$$

$$\mathrm{E}[x] = \int xp(x)dx \qquad \text{if } x \text{ is continuous} \tag{1.18}$$

Giả sử f là một hàm số trả về một giá trị với mỗi giá trị x^* của biến ngẫu nhiên x. Khi đó, nếu x là biến ngẫu nhiên rời rạc, ta sẽ có:

$$\mathrm{E}[f(x)] = \sum_x f(x)p(x) \tag{1.19}$$

Công thức cho biến ngẫu nhiên liên tục cũng được viết tương tự.

Với joint probability:

$$\mathrm{E}[f(x, y)] = \sum_{x,y} f(x, y)p(x, y)dxdy \tag{1.20}$$

Có 3 quy tắc cần nhớ về kỳ vọng:

1. Kỳ vọng của một hằng số theo một biến ngẫu nhiên x bất kỳ bằng chính hằng số đó:

$$\mathrm{E}[\alpha] = \alpha \tag{1.21}$$

2. Kỳ vọng có tính chất tuyến tính:

$$\begin{aligned} \mathrm{E}[\alpha x] &= \alpha \mathrm{E}[x] && (1.22) \\ \mathrm{E}[f(x) + g(x)] &= \mathrm{E}[f(x)] + \mathrm{E}[g(x)] && (1.23) \end{aligned}$$

3. Kỳ vọng của tích hai biến ngẫu nhiên bằng tích kỳ vọng của hai biến đó **nếu hai biến ngẫu nhiên đó là độc lập**. Điều ngược lại không đúng:

$$\mathrm{E}[f(x)g(y)] = \mathrm{E}[f(x)]\mathrm{E}[g(y)] \tag{1.24}$$

1.1.7 Một vài phân phối xác suất thường gặp

Phân phối Bernouli

Phân phối Bernoulli (Bernouli distribution) là một phân bố rời rạc mô tả biến ngẫu nhiên nhị phân: nó mô tả trường hợp khi đầu ra chỉ nhận một trong hai giá trị $x \in \{0, 1\}$. Hai giá trị này có thể là *head* và *tail* khi tung đồng xu; có thể là *fraud transaction* và *normal transaction* trong bài toán xác định giao dịch lừa đảo trong tín dụng; có thể là *người* và *không phải người* trong bài toán tìm xem trong một bức ảnh có người hay không.

Bernoulli distribution được mô tả bằng một tham số $\lambda \in [0, 1]$ và là xác suất để $x = 1$. Phân bố của mỗi đầu ra sẽ là:

$$p(x = 1) = \lambda, \quad p(x = 0) = 1 - p(x = 1) = 1 - \lambda \tag{1.25}$$

Hai đẳng thức này thường được viết gọn lại:

$$p(x) = \lambda^x(1-\lambda)^{1-x} \tag{1.26}$$

với giả định rằng $0^0 = 1$.

Bernoulli distribution được ký hiệu ngắn gọn dưới dạng:

$$p(x) = \text{Bern}_x[\lambda]$$

Phân phối tổng quát của Bernouli (Categorical distribution)

Cũng là biến ngẫu nhiên rời rạc, nhưng trong hầu hết các trường hợp, đầu ra có thể là một trong nhiều hơn hai giá trị khác nhau. Ví dụ, một bức ảnh có thể chứa một chiếc xe, một người, hoặc một con mèo. Khi đó, ta dùng phân bố tổng quát của Bernoulli distribution và được gọi là *Categorical distribution*. Các đầu ra được mô tả bởi 1 phần tử trong tập $\{1, 2, \ldots, K\}$.

Nếu có K đầu ra có thể đạt được, Categorical distribution sẽ được mô tả bởi K tham số, viết dưới dạng vector: $\lambda = [\lambda_1, \lambda_2, \ldots, \lambda_K]$ với các λ_k không âm và có tổng bằng 1. Mỗi giá trị λ_k thể hiện xác suất để đầu ra nhận giá trị k:

$$p(x = k) = \lambda_k$$

Viết gọn lại:

$$p(x) = \text{Cat}_x[\lambda]$$

Biểu diễn theo cách khác, ta có thể coi như đầu ra là một vector ở dạng *one-hot* vector, tức $\mathbf{x} \in \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_K\}$ với $\mathbf{e}_k$ là vector đơn vị thứ k, tức tất cả các phần tử bằng 0, trừ phần tử thứ k bằng 1. Khi đó, ta sẽ có:

$$p(\mathbf{x} = \mathbf{e}_k) = \prod_{j=1}^{K} \lambda_j^{x_j} = \lambda_k \tag{1.27}$$

Cách viết này được sử dụng rất nhiều trong Machine Learning.

Phân phối chuẩn một biến (Univariate normal distribution)

Phân phối chuẩn 1 biến (univariate normal hoặc Gaussian distribution) được định nghĩa trên các biến liên tục nhận giá trị $x \in (-\infty, \infty)$.

Phân phối này được mô tả bởi hai tham số: *mean* μ và *variance* σ^2. Giá trị μ có thể là bất kỳ số thực nào, thể hiện vị trí của *peak*, tức tại đó mà hàm mật độ xác suất đạt giá trị cao nhất. Giá trị σ^2 là một giá trị dương, với σ thể hiện *độ rộng* của phân bố này. σ lớn chứng tỏ khoảng giá trị đầu ra biến đổi mạnh, và ngược lại.

Hàm mật độ xác suất của phân phối này được định nghĩa là:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{1.28}$$

Dạng gọn hơn:

$$p(x) = \text{Norm}_x[\mu, \sigma^2] \tag{1.29}$$

1.2 Giới thiệu Machine Learning

1.2.1 Khái niệm

Trên Wikipedia tiếng Anh họ có định nghĩa về machine learning như sau:

> "**Machine learning** is a subset of artificial intelligence in the field of computer science that often uses statistical techniques to give computers the ability to 'learn' (i.e., progressively improve performance on a specific task) with data, without being explicitly programmed" [3]

Chúng ta có thể hiểu đơn giản định nghĩa ở trên như sau:

Machine learning là một lĩnh vực nhỏ của trí tuệ nhân tạo (*Artificial Intelligence - AI*) trong khoa học máy tính, nó thường sử dụng các kĩ thuật thống kê để máy tính có khả năng

'học' với dữ liệu, mà không cần phải lập trình cụ thể.

Tên gọi *machine learning* được đặt bởi Arthur Samuel [3] năm 1959. Phát triển từ nghiên cứu về nhận dạng mẫu (*pattern recognition*) và lý thuyết học tính toán (*Computational learning theory*) trong trí tuệ nhân tạo, machine learning nghiên cứu và xây dựng các thuật toán có thể học hỏi và dự đoán theo hướng dữ liệu.

Machine learning có mối quan hệ rất mật thiết đối với thống kê (*statistics*). 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.

1.2.2 Phân nhóm thuật toán cơ bản

Trong phần này, chúng tôi trình bày các nhóm thuật toán trong machine learning theo phương thức học. Gồm có 4 nhóm cơ bản sau: học có giám sát (*Supervise learning*), học không giám sát (*Unsupervised learning*), học bán giám sát (*Semi-supervised lerning*) và học củng cố (*Reinforcement learning*).

Học có giám sát (*Supervise learning*)

Supervised learning là thuật toán dự đoán đầu ra (*outcome*) của một dữ liệu mới 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

[3] Arthur Lee Samuel (1901 – 1990) là một nhà tiên phong người Mỹ trong lĩnh vực trò chơi máy tính và trí tuệ nhân tạo

Learning.

Supervised learning là khi chúng ra có một tập hợp biến đầu vào (*data*) $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ và một tập hợp nhãn (*label*) tương ứng $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N\}$, trong đó $\mathbf{x}_i, \mathbf{y}_i$ là các vector.

Các cặp dữ liệu biết trước $(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{X} \times \mathcal{Y}$ được gọi là tập dữ liệu huấn luyện (*training data*). Từ tập traing data này, chúng ta cần tạo ra **một hàm số ánh xạ** mỗi phần tử từ tập $\mathcal{X}$ sang một phần tử (xấp xỉ) tương ứng của tập $\mathcal{Y}$:

$$\mathbf{y}_i \approx f(\mathbf{x}_i), \quad \forall i = 1, 2, \ldots, N$$

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

Ví dụ 1.2. 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.

Thuật toán supervised learning còn được tiếp tục chia nhỏ ra thành hai loại chính tùy thuộc vào label của dữ liệu:

- **Phân loại (Classification)**

  Một bài toán được gọi là *classification* nếu các label của input data được chia thành một số hữu hạn nhóm. Ví dụ 1.2 thuộc loại này.

- **Hồi quy (Regression)**

  Nếu label không được chia thành các nhóm mà là một giá trị thực cụ thể.

  Ví dụ 1.3. Một căn nhà rộng x m^2, có y phòng ngủ và cách trung tâm thành phố z km sẽ có giá là bao nhiêu?

Học không giám sát (*Unsupervised learning*)

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.

Unsupervised learning là khi chúng ta chỉ có dữ liệu vào $\mathcal{X}$ mà không biết nhãn $\mathcal{Y}$ 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.

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

- **Phân nhóm (clustering)**

  Một bài toán phân nhóm toàn bộ dữ liệu $\mathcal{X}$ 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.

Học bán giám sát (*Semi-Supervised learning*)

Các bài toán khi chúng ta có một lượng lớn dữ liệu $\mathcal{X}$ 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.

Học củng cố (*Reinforcement learning*)

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.

Ví dụ 1.4. 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* [4]. 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

[4]Deep Blue là một máy tính chơi cờ vua do IBM phát triển. Deep Blue đã chiến thắng trận đấu đầu tiên của mình với một nhà vô địch thế giới vào ngày 10 tháng 2 năm 1996

chơi để tìm ra các nước đi mới tối ưu hơn. Thuật toán trong phần tự chơi này được xếp vào loại Reinforcement learning.

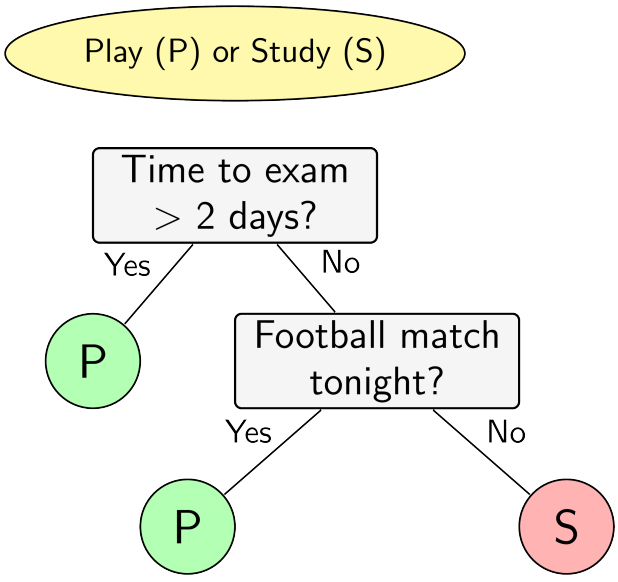
1.3 Thuật toán Decision Tree

1.3.1 Giới thiệu

Sắp đến kỳ thi, một cậu sinh viên tự đặt ra quy tắc *học hay chơi* của mình như sau:

- Nếu còn nhiều hơn hai ngày tới ngày thi, cậu sẽ đi chơi.
- Nếu còn không quá hai ngày và đêm hôm đó có một trận bóng đá, cậu sẽ sang nhà bạn chơi và cùng xem bóng đêm đó.
- Cậu sẽ chỉ học trong các trường hợp còn lại.

Việc ra quyết định của cậu sinh viên này có thể được mô tả trên sơ đồ trong hình 1.2.



Hình 1.2: Một ví dụ về việc đưa ra các quyết định dựa trên câu hỏi

Hình ellipse nền vàng thể hiện quyết định cần được đưa ra. Quyết định này phụ thuộc vào các câu trả lời của các câu hỏi trong các ô hình chữ nhật màu xám. Dựa trên các câu trả lời, quyết định cuối cùng được cho trong các hình tròn màu lục (chơi) và đỏ (học).

Việc quan sát, suy nghĩ và ra các quyết định của con người thường được bắt đầu từ các câu hỏi. Machine learning cũng có một mô hình ra quyết định dựa trên các câu hỏi. Mô hình này có tên là *cây quyết định (decision tree)*.

Trong **decision tree**, các ô màu xám, lục, đỏ trên hình 1.2 được gọi là các node. Các node thể hiện đầu ra (màu lục và đỏ) được gọi là *node lá* (*leaf node* hoặc *terminal node*). Các node thể hiện câu hỏi là các *non-leaf node*. Non-leaf node trên cùng (câu hỏi đầu tiên) được gọi là node gốc (*root node*). Các non-leaf node thường có hai hoặc nhiều node con (*child node*). Các child node này có thể là một leaf node hoặc một non-leaf node khác. Các child node có cùng bố mẹ được gọi là *sibling node*. Nếu tất cả các non-leaf node chỉ có hai child node, ta nói rằng đó là một *binary decision tree* (cây quyết định nhị phân). Các câu hỏi trong binary decision tree đều có thể đưa được về dạng câu hỏi đúng hay sai. Các decision tree mà một leaf node có nhiều child node cũng có thể được đưa về dạng một binary decision tree. Điều này có thể đạt được vì hầu hết các câu hỏi đều có thể được đưa về dạng câu hỏi đúng sai.

Ví dụ, ta có thể xác định được tuổi của một người dựa trên nhiều câu hỏi đúng sai dạng: tuổi của bạn lớn hơn x đúng không? (Đây chính là thuật toán *tìm kiếm nhị phân – binary search*.)

Decision tree là một mô hình *supervised learning*, có thể được áp dụng vào cả hai bài toán *classification và regression*. Việc xây dựng một decision tree trên dữ liệu huấn luyện cho trước là việc đi xác định các câu hỏi và thứ tự của chúng. Một điểm đáng lưu ý của decision tree là nó có thể làm việc với các đặc trưng (trong các tài liệu về decision tree, các đặc trưng thường được gọi là thuộc tính – *attribute*) dạng *categorical*, thường là rời rạc và không có thứ tự. Ví dụ, mưa, nắng hay xanh, đỏ, v.v. Decision tree cũng làm việc với dữ liệu có vector đặc trưng bao gồm cả thuộc tính dạng categorical và liên tục (numeric). Một điểm đáng lưu ý nữa là decision tree ít yêu cầu việc chuẩn hoá dữ liệu.

1.3.2 Phân loại

Có 3 loại decision trees phổ biến sau:

- **ID3 (Iterative Dichotomiser 3)** - Tạo cây nhiều chiều, tìm cho mỗi node một đặt tính phân loại sao cho đặt tính này có giá trị "information gain" lớn nhất. Cây được phát triển tới mức tối đa kích thước. Sau đó áp dụng phương thức cắt tỉa cành để xử lý những dữ liệu chưa nhìn thấy.
- **C4.5** - Kế thừa từ ID3 nhưng loại bỏ hạn chế về việc chỉ sử dụng đặc tính có giá trị phân loại bằng cách tự động định nghĩa một thuộc tính rời rạc. Dùng để phân chia những thuộc tính liên tục thành những tập rời rạc.
- **CART (Classification and Regression Trees)** - Tương tự như C4.5, nhưng nó hỗ trợ thêm đối tượng dự đoán là giá trị số (*Regression*). Cấu trúc CART dạng cây nhị phân, mỗi node sử dụng một ngưỡng để đạt được "information gian" lớn nhất.

Hình 1.3 so sánh giữa các loại thuật toán decision tree.

	Splitting Criteria	Attribute type	Missing values	Pruning Strategy	Outlier Detection
ID3	Information Gain	Handles only Categorical value	Do not handle missing values.	No pruning is done	Susceptible to outliers
CART	Towing Criteria	Handles both Categorical & Numeric value	Handle missing values.	Cost-Complexity pruning is used	Can handle Outliers
C4.5	Gain Ratio	Handles both Categorical & Numeric value	Handle missing values.	Error Based pruning is used	Susceptible to outliers

Hình 1.3: So sánh các thuật toán Decision Trees

1.3.3 Ưu và nhược điểm của thuật toán

Tùy vào loại Decision tree sử dụng mà ta có ưu nhược điểm riêng. Nhưng nhìn chung thuật toán có những ưu nhược điểm chung như sau:

Về ưu điểm

- Decision tree thường mô phỏng cách suy nghĩ con người. Vì vậy nó đơn giản để hiểu và diễn giải dữ liệu.
- Giúp ta nhìn thấy được sự logic của dữ liệu (không như thuật toán phần laoij SVM, KNN . . .)
- Có khả năng chọn được những features tốt nhất.
- Phân loại dữ liệu không cần tính toán phức tạp.
- Giải quyết vấn đề nhiễu và thiếu dữ liệu.
- Có khả năng xử lý dữ liệu có biến liên tục và rời rạc.

Về nhược điểm

- Tỉ lệ tính toán tăng theo hàm số mũ còn vấn đề ngày càng lớn hơn.
- Dễ bị vấn đề overfitting và high bias khi tập dữ liệu huấn luyện nhỏ.

Trong bài báo cáo này chúng tôi sử dụng loại **CART**. Do tính đơn giản, dể tiếp cận của nó, cũng như những giá trị feature mà ta sử dụng là kiểu dữ liệu biến liên tục không phải phân loại nên không dùng **ID3** được. Và đây là loại decision tree được thư viện *scikit-learn* chọn sử dụng.

1.3.4 Làm sạch dữ liệu

1.3.5 Quá trình xây dựng cây

1.4 Một số lỗ hổng tấn công web phổ biến

1.4.1 SQL Injection

1.4.2 Cross-Site Scripting (XSS)

Chương 2

VẬN DỤNG THUẬT TOÁN VÀO PHÂN TÍCH TẬP DỮ LIỆU

2.1 Tập dữ liệu được sử dụng để training

2.2 Thực hiện áp dụng thuật toán vào tập dữ liệu

Tài liệu tham khảo

[1] S.J.D. Prince. *Computer Vision: Models Learning and Inference*. Cambridge University Press, 2012.

[2] Vũ Hữu Tiệp. *Ôn tập Xác Suất cho Machine Learning*. URL: https://goo.gl/BUJH6b.

[3] Wikipedia. *Machine Learning*. URL: https://en.wikipedia.org/wiki/Machine_learning.