**Support Vector Machine**

**https://www.youtube.com/watch?time\_continue=42&v=3liCbRZPrZA**

**Support Vector Machine (SVM) là gì?**

**SVM** là model học có giám sát (tức ta cần labeled cho dataset).

Ví dụ: Tôi có một doanh nghiệp và tôi nhận được rất nhiều email từ khách hàng mỗi ngày. Một số email này là email khiếu nại và cần được trả lời rất nhanh. Tôi muốn có cách xác định chúng nhanh chóng để tôi ưu tiên trả lời các email này.

Ta sẽ dùng ML để phân loại các email bằng thứ tự sau:

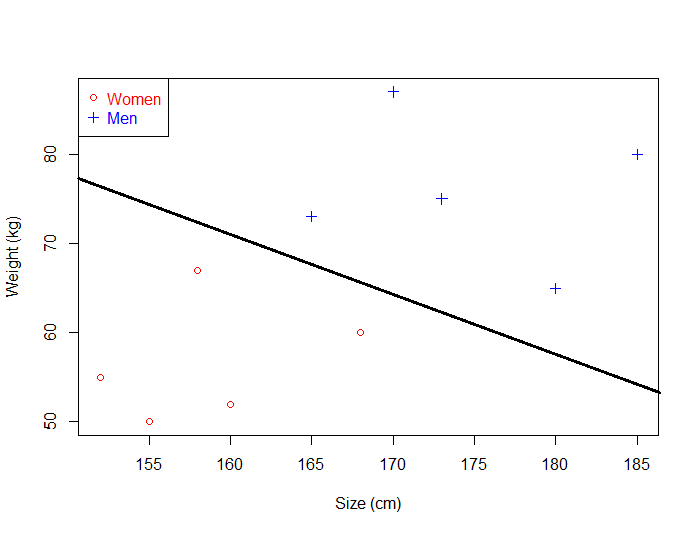
* **Bước 1:** Tôi cần rất nhiều email mẫu làm sample, càng nhiều càng tốt
* **Bước 2:** Tôi sẽ đọc title của từng email và tiến hành gán nhãn ra 2 loại như *“Email phàn nàn”* hoặc *“Email không phàn nàn”*
* **Bước 3:** Tôi sẽ train model trên tập dataset này
* **Bước 4:** Tôi sẽ đánh giá model (ví dụ như dùng cross validation)
* **Bước 5:** Tôi sẽ dùng chính model này để dự đoán liệu đây là email phàn nàn hay không phàn nàn

**SVM học từ mô hình tuyến tính**

Bây giờ chúng ta đã thấy trong ví dụ trước của chúng ta rằng ở **Bước 3** một thuật toán học giám sát như SVM được huấn luyện với dữ liệu có nhãn. Nhưng nó được huấn luyện như thế nào, nó đã học những gì?

**SVM** đã học từ mô hình tuyến tính ***(Linear Model)***

Mô hình tuyến tính là gì? Nói một cách đơn giản, nó là một đường thẳng. Còn nói một cách chuyên môn thì nó đường gọi là ***Hyperplane***, đó là cách gọi tổng quát của các "mặt phẳng" chiều trong một không gian chiều.



*Mục tiêu của SVM là tìm một đường thẳng để phân tách dữ liệu*

Nếu nó chỉ là một đường thẳng, tại sao ta lại nói về mô hình tuyến tính?

Bởi vì bạn không thể học bất cứ cái gì từ một đường thẳng.

Vậy, thay vì:

* Ta giả sử rằng dữ liệu chúng ta muốn phân loại, được học từ một đường thẳng.
* Ta biết rằng một đường thẳng có thể biểu diễn bởi một phương trình(đây cũng chính là model của chúng ta)
* Ta biết rằng ta có thể có vô vàn các dòng như thế bằng cách thay đổi và .
* Ta dùng thuật toán để quyết định giá trị nào của và đưa ra đường một đường thẳng tốt nhất để phân loại dữ liệu

**SVM** là một trong những thuật toán như vậy.

**Mô hình hay thuật toán?**

Khi bắt đầu bài viết tôi đã nói **SVM** là một mô hình học tập có giám sát và bây giờ tôi nói đó là một thuật toán. WTF? Thuật toán thuật ngữ thường được sử dụng một cách lỏng lẻo. Ví dụ, đôi khi bạn sẽ đọc rằng **SVM** là một thuật toán học được giám sát. Điều này không đúng nếu bạn xem xét một thuật toán là một tập hợp các hành động để thực hiện để có được một kết quả cụ thể. Tối ưu hóa tối thiểu tuần tự ***(Sequential minimal optimization)*** là thuật toán được sử dụng nhiều nhất để đào tạo **SVM**, nhưng bạn có thể đào tạo SVM bằng một thuật toán khác như ***Coordinate descent***. Tuy nhiên, hầu hết mọi người không quan tâm đến chi tiết như thế này, vì vậy tôi sẽ đơn giản hóa và nói rằng chúng tôi sử dụng thuật toán SVM '' (không nói chi tiết cái nào chúng tôi sử dụng).

**SVM hay SVMs?**

Thỉnh thoảng, bạn sẽ thấy mọi người nói về SVM và đôi khi về SVMs. Tại sao vậy? đây là định nghĩa từ Wikipedia

*In machine learning, support vector machines (SVMs) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis.*

Okay, bây giờ chúng ta sẽ discover một vài models thuộc họ **SVM**

**SVMs – Support Vector Machine**

Wikipedia nói rằng ta có thể dùng **SVM** trong bài toán: ***classification*** và ***regression****. Cụ thể:*

* **SVM *(Support Vector Machine)*** được dùng cho bài toán ***classification***
* **SVR *(Support Vector Regression)*** được dùng cho bài toán ***regression***

**Classification**

Năm 1957, một mô hình tuyến tính đơn giản được gọi là ***Perceptron*** được phát minh bởi ***Frank Rosenblatt*** để phân loại (thực tế là một trong những khối xây dựng của các mạng nơron đơn giản cũng được gọi là ***Multilayer Perceptron***).

Để biết thêm về ***Perceptron*** là gì đọc bài này nhé:

*https://www.svm-tutorial.com/2017/02/svms-overview-support-vector-machines/* (bạn này viết cực hay và dễ hiểu).

Sau đó, vào năm 1992, ***Vapnik*** và cộng sự đã có ý tưởng áp dụng những gì được gọi là **Kernel Trick**, cho phép sử dụng **SVM** để phân loại dữ liệu không thể tách rời tuyến tính.

Paper về bài báo đó:

*http://www.eric-kim.net/eric-kim-net/posts/1/kernel\_trick\_blog\_ekim\_12\_20\_2017.pdf*

Vậy khi nói đến ***Classification,*** ta có 4 loại **SVM** khác nhau:

* Maximal Margin Classifier
* Kernel Trick
* Soft-margin version
* Soft-margin kernelized version (là sự tổng hợp của 1, 2 và 3)

***Soft-margin kernelized version*** được sử dụng hầu hết thời gian. Đó là lý do tại sao các **SVM** có thể khó hiểu lúc đầu, bởi vì chúng được tạo thành từ nhiều phần đi kèm theo.

Đó là lý do tại sao khi bạn sử dụng ngôn ngữ lập trình, bạn thường được yêu cầu hỏi **Kernel** nào bạn muốn sử dụng (***kernel trick***) và giá trị của tham số **C** bạn muốn sử dụng (vì nó ảnh hưởng tới kết quả của ***the software margin****.*

**Regression**

Năm 1996, Vapnik và cộng sự đã đề xuất một phiên bản **SVM** để thực hiện hồi quy thay vì phân loại. Nó được gọi là ***Support Vector Regression (SVR).*** Giống như phân loại **SVM**, mô hình này bao gồm tham số **C** và ***kernel trick***.

**Thuật ngữ margin trong SVM có nghĩa là gì**?

Margin là khoảng cách giữa siêu phẳng đến 2 điểm dữ liệu gần nhất tương ứng với các phân lớp. Trong ví dụ quả bóng và cái bàn, đó là khoảng cách giữa cây que và hai quả bóng xanh và đỏ gần nó nhất.

Điểm mấu chốt ở đây đó là SVM cố gắng maximize margin này, từ đó thu được một siêu phẳng tạo khoảng cách xa nhất so với 2 quả bóng xanh và đỏ. Nhờ vậy, SVM có thể giảm thiểu việc phân lớp sai (misclassification) đối với điểm dữ liệu mới đưa vào.