**Linear Regression (Hồi Quy Tuyến Tính)**

Giải thích: Linear hay tuyến tính hiểu một cách đơn giản là thẳng, phẳng. Trong không gian hai chiều, một hàm số được gọi là tuyến tính nếu đồ thị của nó có dạng một đường thẳng. Trong không gian ba chiều, một hàm số được goi là tuyến tính nếu đồ thị của nó có dạng một mặt phẳng. Trong không gian nhiều hơn 3 chiều, khái niệm mặt phẳng không còn phù hợp nữa, thay vào đó, một khái niệm khác ra đời được gọi là siêu mặt phẳng (*hyperplane*). Các hàm số tuyến tính là các hàm đơn giản nhất, vì chúng thuận tiện trong việc hình dung và tính toán. Chúng ta sẽ được thấy trong các bài viết sau, tuyến tính rất quan trọng và hữu ích trong các bài toán Machine Learning.

**Giới thiệu:**

Một căn nhà rộng  , có  phòng ngủ và cách trung tâm thành phố  **km** có giá là bao nhiêu. Giả sử chúng ta đã có số liệu thống kê từ 1000 căn nhà trong thành phố đó, liệu rằng khi có một căn nhà mới với các thông số về diện tích, số phòng ngủ và khoảng cách tới trung tâm, chúng ta có thể dự đoán được giá của căn nhà đó không? Nếu có thì hàm dự đoán  sẽ có dạng như thế nào. Ở đây  là một vector hàng chứa thông tin input,  là một số vô hướng (scalar) biểu diễn output (tức giá của căn nhà trong ví dụ này).

Một cách đơn giản nhất, chúng ta có thể thấy rằng:

* Dện tích nhà càng lớn thì giá nhà càng cao;
* Số lượng phòng ngủ càng lớn thì giá nhà càng cao
* Càng xa trung tâm thì giá nhà càng giảm.

Một hàm số đơn giản nhất có thể mô tả mối quan hệ giữa giá nhà và 3 đại lượng đầu vào là:

Trong đó,  là các hằng số, còn được gọi là **bias**/**intercept\_ tức hệ số tự do**. Mối quan hệ  bên trên là một mối quan hệ tuyến tính (*linear*). Bài toán chúng ta đang làm là một bài toán thuộc loại regression. Bài toán đi tìm các hệ số tối ưu chính vì vậy được gọi là bài toán Linear Regression.

**Dạng của Linear Regression**

Trong đó:

* **w = {}** là vector (cột) cần phải tối ưu.
* là vector (hàng) dữ liệu đầu vào, số **1** được đặt ở đầu để phép tính đơn giản và thuận tiện hơn cho việc tính toán.

**Sai số dự đoán:**

Chúng ta mong muốn rằng sự sai khác  giữa giá trị thực  và giá trị dự đoán  (đọc là y hat trong tiếng Anh) là nhỏ nhất. Nói cách khác, chúng ta muốn giá trị sau đây càng nhỏ càng tốt:

**=**

**Hàm mất mát (cost function/loss funtion)**

Điều tương tự xảy ra với tất cả các cặp (input, outcome) với **N** là số dữ liệu quan sát được. Điều chúng ta muốn, tổng sai số là nhỏ nhất, tương đương với việc tìm **w** để hàm số sau đạt giá trị nhỏ nhất:

Hàm số  được gọi là **hàm mất mát** (cost function/loss function) của bài toán Linear Regression. Chúng ta luôn mong muốn rằng sự mất mát (sai số) là nhỏ nhất, điều đó đồng nghĩa với việc tìm vector hệ số  sao cho giá trị của hàm mất mát này càng nhỏ càng tốt. Giá trị của  làm cho hàm mất mát đạt giá trị nhỏ nhất được gọi là điểm tối ưu (optimal point), ký hiệu:

Trước khi đi tìm lời giải, chúng ta đơn giản hóa phép toán trong phương trình hàm mất mát . Đặt là một vector cột chứa tất cả các output của training data. là ma trận dữ liệu đầu vào (mở rộng) mà mỗi hàng của nó là một điểm dữ liệu. Khi đó hàm số mất mát  được viết dưới dạng ma trận đơn giản hơn:

Với là Euclidean norm (chuẩn Euclid, hay khoảng cách Euclid), nói cách khác là tổng bình phương mỗi phần tử của vector **z**. Tới đây, ta đã có một mạng đơn giản của hàm mất mát được viết như phương trình **.**

**Nghiệm cho bài toán Linear Regression**