

TRƯỜNG ĐẠI HỌC THỦY LỢI

**KHOA CÔNG NGHỆ THÔNG TIN**

BÀI TẬP LỚN

HỌC PHẦN: HỌC MÁY

# ĐỀ TÀI: **Paris Housing Price Prediction**

Giáo viên hướng dẫn: Nguyễn Huy Đức

Sinh viên/nhóm sinh viên thực hiện:

1. Nguyễn Gia Bảo, lớp 62TH-VA
2. Lê Đạt Anh, lớp 62TH-VA
3. Nguyễn Đức Anh, lớp 62TH-VA

**Hà Nội, năm 2022**

## Phần 1: Tổng quan

# giới thiệu về học máy

**-** Học máy (Machine Learning): nghiên cứu và xây dựng các kĩ thuật cho phép các

hệ thống “học” tự động từ dữ liệu để giải quyết những vấn đề cụ thể

- Học máy có giám sát: Mô hình học có giám sát là mô hình học trên dữ liệu có dán nhãn, tức là mục tiêu của bài toán machine learning cần học đã được gán nhãn sẵn trong dữ liệu huấn luyện

\*ưu điểm:

- Học tập có giám sát cho phép thu thập dữ liệu và tạo ra dữ liệu đầu ra từ những kinh nghiệm trước đó.

- Giúp tối ưu hóa các tiêu chí hiệu suất với sự trợ giúp của kinh nghiệm.

- ML có giám sát giúp giải quyết nhiều loại vấn đề tính toán trong thế giới thực.

**\*nhược điểm:**

- Phân loại dữ liệu lớn có thể là một thách thức.

- Đào tạo cho việc học có giám sát cần rất nhiều thời gian tính toán, vì vậy, nó đòi hỏi rất nhiều thời gian.

- Học máy không giám sát: Học không giám sát là một lớp mô hình học sử dụng một thuật toán để mô tả hoặc trích xuất ra các mối quan hệ tiềm ẩn trong dữ liệu. Khác với học có giám sát, học không giám sát chỉ thực thi trên dữ liệu đầu vào không cần các thuộc tính nhãn, hoặc mục tiêu của việc học.

# \*ưu điểm:

- Nó có thể thấy những gì tâm trí con người không thể hình dung.

- Nó được sử dụng để đào các mẫu ẩn có tầm quan trọng tối đa trong ngành và có các ứng dụng rộng rãi trong thời gian thực.

- Có sự phức tạp ít hơn so với nhiệm vụ học tập có giám sát. Ở đây, không ai được yêu cầu giải thích các nhãn liên quan và do đó nó giữ độ phức tạp ít hơn.

- Nó dễ dàng hơn một cách hợp lý để có được dữ liệu không nhãn.

**\*nhược điểm:**

- Nó tốn kém hơn vì nó có thể đòi hỏi sự can thiệp của con người để hiểu các mô hình và tương quan chúng với kiến thức miền

- Không phải lúc nào cũng chắc chắn rằng các kết quả thu được sẽ hữu ích vì không có thước đo nhãn hoặc đầu ra để xác nhận tính hữu dụng của nó.

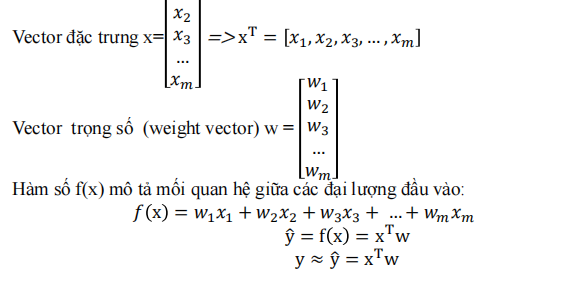
- Kết quả thường có độ chính xác thấp hơn

### *Mô tả bài toán*

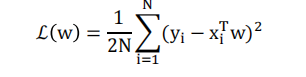
* + Tên bài toán: Paris Housing Price Prediction (Dự đoán giá nhà ở Paris)
  + Input: Dữ liệu sử dụng để dự đoán
  + Ouput: Giá dự đoán của ngôi nhà dựa vào dữ liệu trên

### *Phương pháp học máy*

*\** Linear Regression:



+)hàm mất mát:



+)nghiệm của bài toán:

- Tìm giá trị tối ưu của w có thể được thực hiện thông qua việc giải phương trình đạo hàm của ℒ w theo w bằng không

-Đạo hàm theo w của hàm ℒ w là:



- Giải phương pháp đạo hàm bằng không:



- Nếu ma trận XXT khả nghịch thì phương trình trên có nghiệm duy nhất là:



- Nếu ma trận XXT không khả nghịch thì nghiệm của phương trình có thể xác

định dựa vào giả nghịch đảo:



\*PCA:

+ Giới thiệu:

Dimensionality Reduction (giảm chiều dữ liệu), là một trong những kỹ thuật quan trọng trong Machine Learning. Các feature vectors trong các bài toán thực tế có thể có số chiều rất lớn, tới vài nghìn. Ngoài ra, số lượng các điểm dữ liệu cũng thường rất lớn. Nếu thực hiện lưu trữ và tính toán trực tiếp trên dữ liệu có số chiều cao này thì sẽ gặp khó khăn cả về việc lưu trữ và tốc độ tính toán. Một phương pháp đơn giản nhất trong các thuật toán Dimensionality Reduction dựa trên một mô hình tuyến tính. Phương pháp này có tên là Principal Component Analysis (PCA), tức Phân tích thành phần chính.

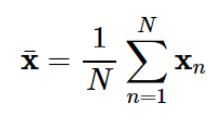
+ Định nghĩa:

-PCA là phương pháp đi tìm một hệ trực chuẩn mới sao cho trong hệ này, các thành phần quan trọng nhất nằm trong K thành phần đầu tiên.

-Để cho đơn giản trong tính toán, PCA sẽ tìm một hệ trực chuẩn để làm cơ sở mới.

+ Các bước thực hiên PCA:

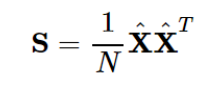
B1: Tính vector kỳ vọng của toàn bộ dữ liệu:



B2: Trừ mỗi điểm dữ liệu đi vector kỳ vọng của toàn bộ dữ liệu:



B3: Tính ma trận hiệp phương sai S:



B4: Tính các trị riêng và vector riêng của ma trận S , sắp xếp chúng theo thứ tự giảm dần của giá trị riêng.

B5: Chọn vector riêng ứng với trị riêng lớn nhất để xây dựng ma trận Uk có

các cột tạo thành một hệ trực giao. vectors này là các thành phần chính.

B6:Chiếu dữ liệu ban đầu đã chuẩn hoá xuống không gian con tìm được. Dữ

liệu mới chính là toạ độ của các điểm dữ liệu trên không gian mới.

+ Làm thế nào để chọn K ( số chiều cho dữ liệu mới) hợp lý:

Có một cách xác định KK là dựa trên việc lượng thông tin muốn giữ lại. Như đã trình bày, PCA còn được gọi là phương pháp tối đa tổng phương sai được giữ lại. Vậy ta có thể coi tổng các phương sai được giữ lại là lượng thông tin được giữ lại. Với phương sai càng lớn, tức dữ liệu có độ phân tán cao, thể hiện lượng thông tin càng lớn.

Trong mọi hệ trục toạ độ, tổng phương sai của dữ liệu là như nhau và bằng tổng các trị riêng của ma trận hiệp phương sai ∑Di=1λi∑i=1Dλi. Thêm nữa, PCA giúp giữ lại lượng thông tin (tổng các phương sai) là: ∑Ki=1λi∑i=1Kλi. Vậy ta có thể coi biểu thức:



là lượng thông tin được giữ lại khi số chiều dữ liệu mới sau PCA là K.

Như vậy, giả sử ta muốn giữ lại 99% dữ liệu, ta chỉ cần chọn K là số tự nhiên nhỏ nhất sao cho rK≥0.99

Khi dữ liệu phân bố quanh một không gian con, các giá trị phương sai lớn nhất ứng với các λi đầu tiên lớn hơn nhiều so với các phương sai còn lại. Khi đó, ta có thể chọn được K khá nhỏ để đạt được rK≥0.99

+ Một số lưu ý trong việc tính toán PCA trong bài toán thực thế:

**Trong mục này, ta sẽ coi như dữ liệu đã được chuẩn hoá, tức đã được trừ đi vector kỳ vọng. Khi đó, ma trận hiệp phương sai sẽ là**  
 

- Số chiều nhiều hơn số điểm dữ liệu:

- Đó là trường hợp D>N, tức ma trận dữ liệu X là một ‘ma trận cao’. Khi đó, số trị riêng khác không của ma trận hiệp phương sai SS sẽ không vượt quá rank của nó, tức không vượt quá N. Vậy ta cần chọn K≤N vì không thể chọn ra được K>N trị riêng khác 0 của một ma trận có rank bằng N.

- Việc tính toán các trị riêng và vector riêng cũng có thể được thực hiện một cách hiệu quả dựa trên các tính chất sau đây:

**Tính chất 1:** Trị riêng của A cũng là trị riêng của kA với kk≠0 bất kỳ. Điều này có thể được suy ra trực tiếp từ định nghĩa của trị riêng và vector riêng.

**Tính chât 2:** Trị riêng của ABAB cũng là trị riêng của BA với A∈Rd1×d2,B∈Rd2×d1 là các ma trận bất kỳ và d1,d2 là các số tự nhiên khác không bất kỳ. Tôi xin không chứng minh quan sát này.

Như vậy, thay vì tìm trị riêng của ma trận hiệp phương sai S∈RD×, ta đi tìm trị riêng của ma trận T=XTX∈RN×N có số chiều nhỏ hơn (vì  <D).

**Tính chất 3:** Giả sử (λ,u) là một cặp trị riêng - vector riêng của T, thế thì (λ,Xu) là một cặp trị riêng - vector riêng của S.

- Chuẩn hóa các vectors riêng:

Việc cuối cùng phải làm là chuẩn hoá các vector riêng tìm được sao cho chúng tạo thành một hệ trực chuẩn. Việc này có thể dựa trên hai điểm sau đây:

**Thứ nhất**, nếu A là một ma trận đối xứng, (λ1,x1),(λ2,x2) là các căp trị riêng - vector riêng của A với λ1≠λ2, thế thì xT1x2=. Nói cách khác, hai vector bất kỳ trong hai không gian riêng khác nhau của một ma trận đối xứng thì trực giao với nhau. Chứng minh cho tính chất này có thể được thấy trong một dòng dưới đây:



Dấu bằng cuối cùng xảy ra vì λ1≠λ2.

**Thứ hai**, với các trị riêng độc lập tìm được trong một không gian riêng, ta có thể dùng Gram-Schmit process để chuẩn hoá chúng về một hệ trực chuẩn.

Kết hợp hai điểm trên, ta có thể thu được các vector riêng tạo thành một hệ trực chuẩn, chính là ma trận UK trong PCA.

* + - Với các bài toán large\_scale:

## Phần 2: Thực nghiệm

### *Mô tả tập dữ liệu của bài toán*

* + Dữ liệu gồm những chiều thông tin gì (mỗi mẫu (vertor) dữ liệu có những thông tin gì), có bao nhiêu mẫu dữ liệu (ít nhất là 100 vector dữ liệu). Mô tả nhãn lớp của dữ liệu.
  + Toàn bộ dữ liệu của các thuộc tính đều dưới dạng số. Các nhãn lớp của tập dữ liệu:
    - * squareMeters – diện tích
      * numberOfRooms – số phòng
      * hasYard – có sân (1 = true, 0 = false)
      * hasPool – có bể bơi (1 = true, 0 = false)
      * floors – số tầng
      * cityCode – code zip của thành phố
      * cityPartRange – chất lượng dân cư xung quanh
      * numPrevOwners – số chủ từng ở ngôi nhà
      * made – thời điểm hoàn tất xây dựng (năm)
      * isNewBuilt – đã được tân trang (1 = true, 0 = false)
      * hasStormProtector – có chống bão (1 = true, 0 = false)
      * basement – diện tích tầng hầm
      * attic – diện tích gác mái
      * garage – điện tích garare
      * hasStorageRoom – có phòng kho (1 = true, 0 = false)
      * hasGuestRoom – có phòng cho khách (1 = true, 0 = false)
      * price – giá trị ngôi nhà
  + Mô tả ma trận dữ liệu (X), nhãn lớp (Y)
    - X =

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 75523 | 3 | 0 | 1 | 63 | 9373 | 3 | 8 | 2005 | 0 | 1 | 4313 | 9005 | 956 | 0 | 7 |
| 80771 | 39 | 1 | 1 | 98 | 39381 | 8 | 6 | 2015 | 1 | 0 | 3653 | 2436 | 128 | 1 | 2 |
| 55712 | 58 | 0 | 1 | 19 | 34457 | 6 | 8 | 2021 | 0 | 0 | 2937 | 8852 | 135 | 1 | 9 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 1440 | 84 | 0 | 0 | 49 | 18412 | 6 | 10 | 1994 | 1 | 0 | 8485 | 2024 | 278 | 1 | 6 |

* + - Y =

|  |
| --- |
| 7559081.5 |
| 8085989.5 |
| 5574642.1 |
| ... |
| 146708.4 |

* + Chia tập dữ liệu thành 2 phần: 70% dùng để huấn luyện mô hình, 30% dùng để kiểm tra.

### *2. Phân tích kết quả của chương trình*

- Tỉ lệ cao nhất khi dùng PCA cho dự đoán đúng là : 56,2666%

## Kết luận:

+ Dùng phương pháp Phân tích thành phần chính(PCA) để lựa chọn tập các thuộc tính tốt nhất cho bài toán.

+ Từ tập training data và test data ban đầu, sử dụng các thành phần chính tốt nhất đã chọn để tạo ra tập training data và test data mới.

+ tìm dữ liệu có tỉ lệ tốt nhất sau 8 lần chia data rồi gán cho mô hinh dự đoán Linear Regression

***3. Mô tả các chức năng của chương trình***

- chương trình hiển thị giao diện để người dùng nhập số liệu cụ thể cho từng thông tin được đưa ra để dự đoán

## Tài liệu tham khảo

* + Các tài liệu được tham khảo trong báo cáo:

+ scikit-learn

+ machine learning cơ bản

+ hồi quy tuyến tính - Wikipedia