ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN MÔN HỌC

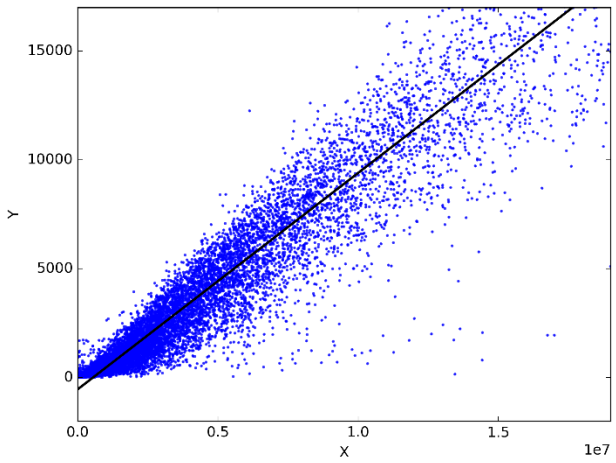
| Đề tài |

LINEAR REGRESSION

| Giảng viên hướng dẫn |

GV. BÙI HUY THÔNG

Môn học: Toán ứng dụng và thống kê



Võ Trần Quang Tuấn - 18127248

Thành phố Hồ Chí Minh – 2020

I. Sinh viên thực hiện.

- Họ và tên: Võ Trần Quang Tuấn.
- MSSV: 18127248
- Lớp: 18CLC2.
- Môn học: Toán ứng dụng và thống kê.

II. Bài toán giải quyết.

1. Mô tả.

- Hồi quy tuyến tính (Linear Regression) là một trong những thuật toán cơ bản sử dụng trong các bài toán dự đoán, ví dụ như dự đoán giá nhà, dự đoán chất lượng sản phẩm, dự đoán kết quả xổ số,… dựa vào các dữ liệu thu thập được từ trước và đưa ra tiên đoán từ những dữ liệu ấy.
- Ý tưởng chính của hồi quy tuyến tính chính là chúng ta tìm một hàm số $f(x)$ xấp xỉ với các điểm dữ liệu dùng để huấn luyện mô hình, từ hàm $f(x)$ này chúng ta có thể thay giá trị các feature (đặc trưng) vào và nhận được giá trị dự đoán của bài toán.

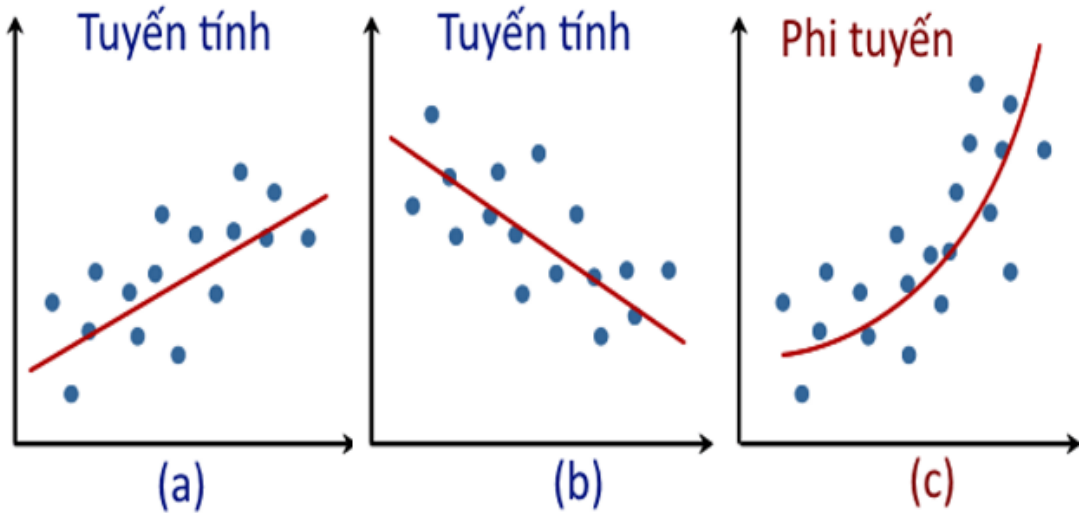


Figure 1: Hồi quy tuyến tính, Hồi quy phi tuyến tính (source: internet).

- Trong đồ án này, chúng ta sẽ đi tìm hiểu hồi quy tuyến tính và cách ứng dụng nó vào bài toán đánh giá chất lượng rượu được lưu trữ trong cơ sở dữ liệu như sau gồm 1199 chai rượu với 11 feature như độ acid, độ cồn, sunfua dioxide,… và điểm đánh giá là giá trị thực nằm trong thang từ 11 đến 10.

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.700	0.00	1.9	0.076	11.0	34	0.99780	3.51	0.56	9.4	5
1	7.8	0.880	0.00	2.6	0.098	25.0	67	0.99680	3.20	0.68	9.8	5
2	7.8	0.760	0.04	2.3	0.092	15.0	54	0.99700	3.26	0.65	9.8	5
3	11.2	0.280	0.56	1.9	0.075	17.0	60	0.99800	3.16	0.58	9.8	6
4	7.4	0.700	0.00	1.9	0.076	11.0	34	0.99780	3.51	0.56	9.4	5
...	...	...	...	...	...	...	...	...	...	...	...	...
1194	7.0	0.745	0.12	1.8	0.114	15.0	64	0.99588	3.22	0.59	9.5	6
1195	6.2	0.430	0.22	1.8	0.078	21.0	56	0.99633	3.52	0.60	9.5	6
1196	7.9	0.580	0.23	2.3	0.076	23.0	94	0.99686	3.21	0.58	9.5	6
1197	7.7	0.570	0.21	1.5	0.069	4.0	9	0.99458	3.16	0.54	9.8	6
1198	7.7	0.260	0.26	2.0	0.052	19.0	77	0.99510	3.15	0.79	10.9	6

1199 rows × 12 columns

Figure 2: Cơ sở dữ liệu rượu, wine.csv.

2. Hướng giải quyết và ý tưởng.

a. Xây dựng mô hình dự đoán chất lượng rượu bằng 11 thuộc tính.

- Như đã giới thiệu về mô hình hồi quy tuyến tính, nhiệm vụ của chúng ta là phải đi tìm một hàm số xấp xỉ giá trị cần dự đoán dựa vào các đặc trưng của dữ liệu.
- Giả sử một điểm dữ liệu mô tả một chai rượu có dạng là một vector 11 chiều $\boldsymbol{x}$ (ứng với 11 đặc trưng cần hồi quy). Label là điểm đánh giá được cho sẵn của 1199 chai rượu là $\boldsymbol{y}$. Ta có:

$$\boldsymbol{x} = [x_0, x_1, \dots, x_{10}]^T \in R^{11}$$

- Gọi $\boldsymbol{\theta}$ là vector tham số huấn luyện tương ứng với model.
- Theo khái niệm về mô hình hồi quy tuyến tính, trong không gian có có dạng là một điểm trong không gian 1 chiều, đường thẳng trong không gian 2 chiều, mặt phẳng trong không gian 3 chiều, siêu phẳng trong không gian n chiều. Để siêu phẳng xấp xỉ được linh hoạt hơn, chúng ta cần thêm một hệ số tự do (bias) vào hàm $f(x)$, vì nếu không có bias thì siêu phẳng chỉ giới hạn đi qua gốc toạ độ. Do đó, mô hình của chúng ta sẽ có dạng:

$$\gamma = f(x) = \theta_0 x_0 + \cdots + \theta_{10} x_{10} + b$$

với b là hệ số tự do bias

- Do đó vector điểm dữ liệu và tham số huấn luyện của chúng ta bây giờ sẽ là:

$$\boldsymbol{x} = [x_0, x_1, \dots, x_{10}, 1]^T \in R^{12}$$
$$\boldsymbol{\theta} = [\theta_0, \theta_1, \dots, \theta_{10}, b] \in R^{12}$$

- Xếp toàn bộ 1199 điểm dữ liệu thành một ma trận $\boldsymbol{A} \in R^{1199\times 12}$ rồi áp dụng phương pháp bình phương tối tiểu (least square) để tìm tham số $\boldsymbol{\theta}$

b. <u>Tìm ra thuộc tính nào ảnh hưởng đến chất lượng rượu nhất.</u>

- Không hẳn trong toàn bộ 11 thuộc tính thì cả 11 thuộc tính đều ảnh hưởng đến chất lượng rượu, do đó chúng ta cần tìm ra thuộc tính ảnh hưởng nhất.
- Chúng ta sẽ tiếp tục sử dụng phương pháp tìm tham số model như câu a (sử dụng bias và bình phương tối tiểu), nhưng có sự khác biệt trong việc chọn dữ liệu huấn luyện.
- Trong phần này sẽ sử dụng phương pháp cross validation test để đánh giá thuộc tính rượu. Chúng ta sẽ chia dataset thành k tập con nhỏ không giao nhau, sau đó chọn bất kì 1 tập con để làm test set, và k-1 tập còn lại làm training set, do đó ứng với mỗi thuộc tính sẽ có k lần huấn huyện.

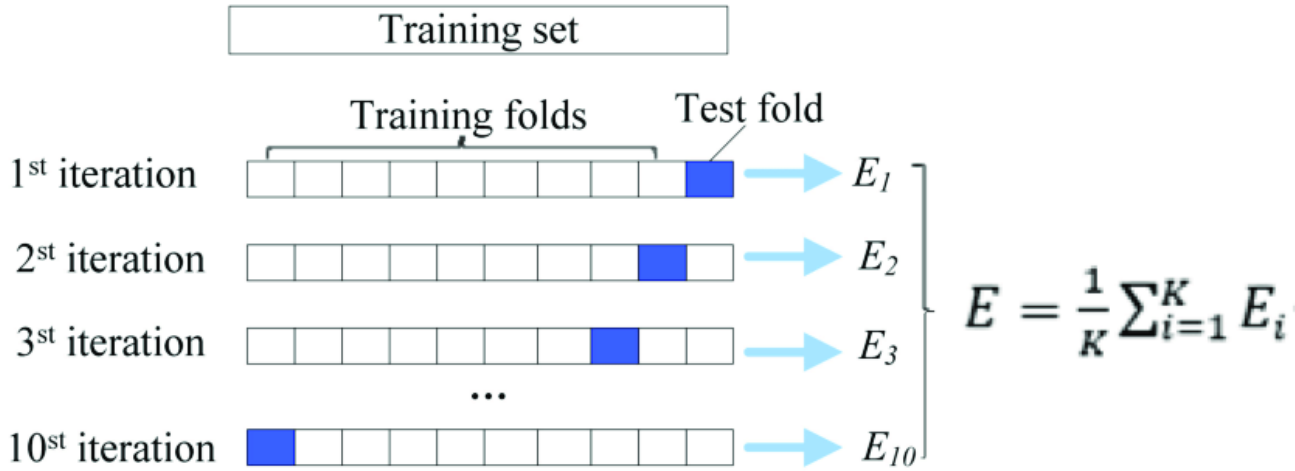


Figure 3: Cross validation (source: internet)

- Ứng với mỗi thuộc tính sẽ có k lần huấn luyện và độ lỗi của thuộc tính đó chính là trung bình độ lỗi của k lần, tham số là tương ứng với lần có độ lỗi nhỏ nhất, tính độ lỗi model bằng trung bình độ lỗi của các điểm dữ liệu trong test set.
- Chọn thuộc tính tốt nhất tương ứng với độ lỗi nhỏ nhất trong 11 thuộc tính.

c. <u>Tự xây dựng mô hình dự đoán cho riêng bạn để đạt được kết quả tốt nhất.</u>

- Vì không hẳn là 11 thuộc tính đều ảnh hưởng đến chất lượng rượu và cũng không hẳn chỉ có 1 thuộc tính ảnh hưởng đến chất lượng rượu. Do đó chúng ta cần tìm ra k thuộc tính đóng vai trò quan trọng trong việc đánh giá chất lượng. Trong phần này, chúng ta sẽ sử dụng phương pháp backward feature elimination.

- Tương tự như câu a và b, chúng ta cũng sẽ thêm bias, dùng phương pháp bình phương tối tiểu và cross validation để tìm ra thuộc tính không quan trọng cần bỏ đi.
- Sau khi quá trình huấn luyện kết thúc (tương ứng với drop toàn bộ 10 thuộc tính), chúng ra sẽ chọn ra lần drop nào có độ lỗi nhỏ nhất, tham số huấn luyện sẽ là tham số của lần drop đó (backward feature elimination).

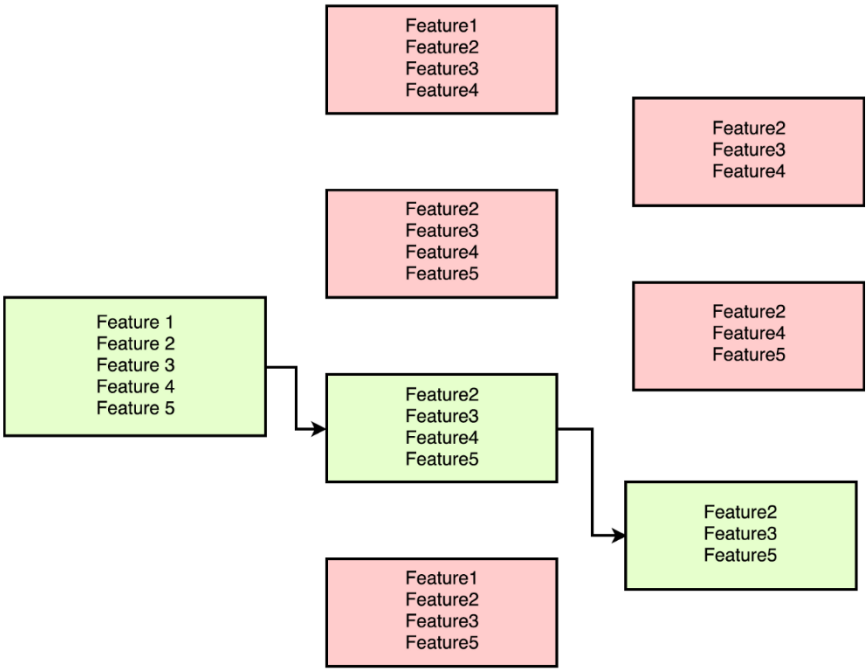


Figure 4: Backward Feature Elimination (source: internet)

III. Các chức năng đã hoàn thành và mô tả hàm.

1. Các chức năng.

- Xây dựng mô hình dự đoán chất lượng rượu bằng 11 thuộc tính.
  Phương pháp sử dụng: thêm bias, bình phương tối tiểu
- Tìm ra thuộc tính nào ảnh hưởng đến chất lượng rượu nhất.
  Phương pháp sử dụng: thêm bias, bình phương tối tiểu, cross validation.
- Tự xây dựng mô hình dự đoán cho riêng bạn để đạt được kết quả tốt nhất.
  Phương pháp sử dụng: thêm bias, bình phương tối tiểu, cross validation, backward feature elimination.

2. Các thư viện sử dụng.

- Thư viện tính toán trên ma trận: numpy
- Thư viện đọc file dữ liệu đầu vào (dạng .csv): pandas
- Thư viện sử dụng cho việc xáo trộn và phân chia dữ liệu cho quá trình huấn luyện model: sklearn.model_selection.

3. Các hàm sử dụng.

a. Xây dựng mô hình dự đoán chất lượng rượu bằng 11 thuộc tính.

- preparing_data(file_name): chuẩn bị dữ liệu cho quá trình huấn luyện, tách dữ liệu thành training set và label set.
  - ➔ Input: tên file input.
  - ➔ Output: training set, label set.
- linear_regression(dataset, label): hồi quy tuyến tính.
  - ➔ Input: tập dữ liệu huấn luyện, label của dữ liệu.
  - ➔ Output: tham số model của quá trình huấn luyện.
- get_errror(dataset, label, para): tính độ lỗi.
  - ➔ Input: tập dữ liệu, nhãn của dữ liệu, tham số model.
  - ➔ Output: độ lỗi của model.

b. Tìm ra thuộc tính nào ảnh hưởng đến chất lượng rượu nhất.

- kFold_validation_set(file_name): chuẩn bị k tập data cho cross validation. Sử dụng thư viện sklearn cho việc tạo k tập validation (from sklearn.model_selection import RepeatedKFold)
  - ➔ Input: tên file input
  - ➔ Output: danh sách chứa k tập của k-1 tập data training, danh sách k tập label của k-1 tập data training, danh sách chứa k tập data testing, danh sách chứa k tập label của tập data testing.
- get_attribute(training_set, testing_set, k): lấy ma trận data tương ứng với thuộc tính k.
  - ➔ Input: tập data huấn luyện, tập data test, thuộc tính k.
  - ➔ Output: tập data huấn luyện và tập data test sau khi bỏ thuộc tính k.
- linear_regression_for_attribute(train_x, train_y, test_x, test_y, k): Tìm độ lỗi và tham số model ứng với từng thuộc tính.
  - ➔ Input: tập data training, label của data training, tập data testing, label của data testing, chỉ số thuộc tính k.
  - ➔ Output: độ lỗi của model, tham số model.
- find_best_attribute(train_x, train_y, test_x, test_y, n_attributes): Tìm thuộc tính ảnh hưởng nhất.
  - ➔ Input: danh sách chứa k tập của k-1 tập data training, danh sách k tập label của k-1 tập data training, danh sách chứa k tập data testing, danh sách chứa k tập label của tập data testing, số thuộc tính.
  - ➔ Output: độ lỗi của model, tham số model, chỉ số của thuộc tính ảnh hưởng nhất.

c. Tự xây dựng mô hình dự đoán của riêng bạn cho kết quả tốt nhất.

- drop_attribute(training_set, testing_set, k): Loại bỏ thuộc tính trong tập training và testing, tham số model ứng với thuộc tính đó sẽ bằng 0.
  - ➔ Input: tập data huấn luyện, tập data test, thuộc tính k.
  - ➔ Output: tập data huấn luyện, tập data test sau khi bỏ thuộc tính k.
- linear_regression_for_validation(train_x, train_y, test_x, test_y, k): hồi quy tuyến tính sau khi bỏ thuộc tính k.
  - ➔ Input: tập data training, label của data training, tập data testing, label của data testing, chỉ số thuộc tính k.
  - ➔ Output: độ lỗi của model, tham số model.
- find_worst_attribute(train_x, train_y, test_x, test_y, n_attributes): tìm thuộc tính tệ nhất (bỏ nó đi nhưng độ lỗi vẫn nhỏ -> chứng tỏ thuộc tính đó ít quan trọng).
  - ➔ Input: tập data training, label của data training, tập data testing, label của data testing, số thuộc tính.
  - ➔ Output: tập data huấn luyện, tập data test sau khi bỏ thuộc tính k, tham số model sau khi bỏ k, độ lỗi, chỉ số k được bỏ đi.
- backward_feature_elimination(train_x, train_y, test_x, test_y, n_attributes): Thực hiện vòng lặp huấn luyện mô hình khi bỏ lần lượt các thuộc tính, đến khi còn 1 thuộc tính thì dừng lại. Chọn cách bỏ nào mà độ lỗi nhỏ nhất.
  - ➔ Input: danh sách chứa k tập của k-1 tập data training, danh sách k tập label của k-1 tập data training, danh sách chứa k tập data testing, danh sách chứa k tập label của tập data testing, số thuộc tính.
  - ➔ Output: Độ lỗi, tham số model sau khi lọc các thuộc tính không quan trọng, danh sách các thuộc tính bị lọc đi.

IV. Kết quả thu được.

1. Kết quả.

a. Xây dựng mô hình dự đoán chất lượng rượu bằng 11 thuộc tính.

- Độ lỗi thu được (error) và tham số model (parameter) tìm được như trên hình.

```
[9]  1 data_points, label = preparing_data("wine.csv")
     2 para_model = linear_regression(data_points, label)
     3 error = get_error(data_points, label, para_model)
     4 print(f"error: {error}")
     5 print(f"parameter: {para_model}")

error: 0.4997901216695821
parameter: [ 4.75247531e-02 -1.06874258e+00 -2.68710829e-01  3.49742662e-02
 -1.59729560e+00  3.48788138e-03 -3.79835506e-03 -3.94690810e+01
 -2.45575908e-01  7.73840794e-01  2.69377496e-01  4.29171625e+01]
```

Figure 5: Kết quả câu a.

b. <u>Tìm ra thuộc tính nào ảnh hưởng đến chất lượng rượu nhất.</u>

- Độ lỗi thu được: 0.56
- Thuộc tính ảnh hưởng nhất (attribute index) có chỉ số là 10, do đó chính là độ cồn (alcohol).

```
[20]  1 training_datasets, training_labels, testing_datasets, testing_labels = kFold_validation_set("wine.csv")
      2 error, para, attribute = find_best_attribute(training_datasets, training_labels, testing_datasets, testing_labels, 11)
      3 print(f"error: {error}")
      4 print(f"parameter: {para}")
      5 print(f"attribute index: {attribute}")

error: 0.5654098361329299
parameter: [0.37131026 1.80838583]
attribute index: 10
```

Figure 6: Kết quả câu b.

c. <u>Tự xây dựng mô hình dự đoán của riêng bạn cho kết quả tốt nhất.</u>

- Độ lỗi thu được (error) và tham số thu được sau khi áp dụng backward feature elimination như trên hình.

```
[41]  1 training_datasets, training_labels, testing_datasets, testing_labels = kFold_validation_set("wine.csv")

[42]  1 error, para, drop = backward_feature_elimination(training_datasets, training_labels, testing_datasets, testing_labels, 11)

      error when training 10 attributes: 0.5058291817677177
      error when training 9 attributes: 0.5053285314501507
      error when training 8 attributes: 0.5058239050912102
      error when training 7 attributes: 0.5065765893914319
      error when training 6 attributes: 0.5071192542647047
      error when training 5 attributes: 0.5084664349519226
      error when training 4 attributes: 0.510514972019384
      error when training 3 attributes: 0.5204380067556061
      error when training 2 attributes: 0.5305594714411308
      error when training 1 attributes: 0.5668273759500324

[43]  1 error

      0.5053285314501507

[44]  1 para

      array([ 5.28264297e-02, -9.57123151e-01,  3.48010568e-02, -1.67051493e+00,
              2.48963557e-03, -3.67709206e-03, -5.38920308e+01,  7.84213235e-01,
              2.48864326e-01,  5.65317617e+01])
```

Figure 7: Kết quả câu c.

- Tham số và độ lỗi trên ứng với việc sau khi bỏ đi thuộc tính thứ 2 là citrid acid và thuộc tính thứ 7 (sau khi bỏ đi thuộc tính thứ 2) là pH.

```
[45]  1 drop

      [2, 7]
```

Figure 8: Thuộc tính bị drop (câu c).

2. Nhận xét.

a. Nhận xét kết quả thu được với hồi quy tuyến tính.

- Với việc đánh giá chất lượng rượu dựa trên toàn bộ 11 thuộc tính dễ gây ra hiện tượng overfitting hay còn được gọi là quá khớp dữ liệu. Overfitting sẽ làm cho model của chúng ra dự đoán sai vì nó quá fit với data huấn luyện, cho nên mất tính tổng quát của bài toán.
- Độ lỗi luôn duy trì ở mức 0.48 -> 0.507. Do đó khó cải thiện model dự đoán này bằng hồi quy tuyến tính.
- Hồi quy tuyến tính rất nhạy cảm với nhiễu.

b. Một số ý tưởng khác.

- Có một số phương pháp tránh overfitting như regularization (ví dụ ridge regression hay LASSO regression), cross validation,…

- LASSO regression có xu hướng cho một số tham số model bằng 0, tương tự ý tưởng của Backward Feature Elimination, đây cũng là một trong những phương pháp feature selection.
- Ridge Regression cũng là linear regression nhưng khác nhau ở chỗ nó thêm một hệ số trong hàm đánh giá để tối ưu độ lỗi là $\lambda \mathbf{I}$ với $\mathbf{I}$ là ma trận đơn vị và λ là một số dương cực nhỏ, mục đích để tránh overfitting.
- Sau khi thực hiện hồi quy cho câu a và c, chúng ta nhận thấy độ lỗi của 2 model không chênh lệch nhau nhiều, do đó chứng tỏ dữ liệu của chúng ta có thể không hoàn toàn tuyến tính, có thể là phi tuyến. Do đó chúng ta có thể điều chỉnh một chút để bài toán trở thành hồi quy phi tuyến, nhưng tìm tham số model vẫn là tuyến tính như sau.

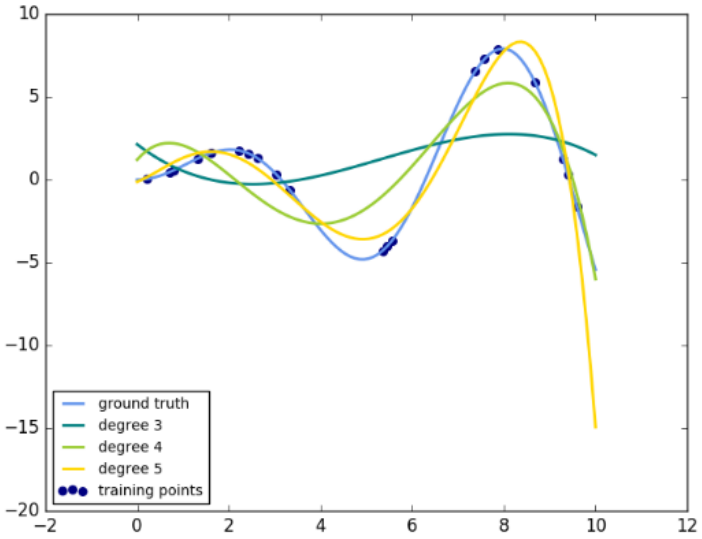


Figure 9: Hồi quy phi tuyến (source: internet).

- Lúc này, hàm số cần tìm để xấp xỉ:

$$f(x) = \theta_0 g_0(x_0) + \theta_1 g_1(x_1) + \cdots + \theta_n g_n(x_n)$$

Ứng với $g_i(x_i)$ là các hàm số phi tuyến như sin(x), cos(x), hàm mũ của x,…
- Việc tìm các hàm số như sin(x), cos(x),… cho model là rất khó khả thi, cho nên chúng ta có thể đơn giản g(x) bằng cách sử dụng các hàm mũ sẽ khả thi hơn và dễ tối ưu hơn vì việc tìm tham số model với các hàm mũ tương ứng với các feature vẫn là tuyến tính. Đây còn được gọi là hồi quy đa thức (polynomial regression).

- Phương pháp: ứng với mỗi vector đặc trưng x, ta tính được vector x_new tương ứng với model hàm mũ rồi sử dụng bình phương tối tiểu để tìm tham số model.
- Nhưng vấn đề ở đây: bậc tối đa của hàm mũ cần sử dụng là bao nhiêu? Tìm bậc đó như thế nào? Vì bậc càng cao thì khả năng xảy ra overfitting là rất lớn.
  ➔ Cross validation có thể giải quyết chuyện này.
- Sử dụng phương pháp cross validation để tìm bậc và đồng thời tìm tham số model, cứ tăng dần bậc của model lên từ từ, và đồng thời quan sát độ lỗi của training set và testing set, tại thời điểm error trong testing set có xu hướng đi lên thì lập tức ngừng quá trình training.
  ➔ Đây cũng chính là ý tưởng của Early Stopping
- Đối với trường hợp dữ liệu quá lớn, số đặc trưng quá nhiều, chúng ta không thể áp dụng tính giả nghịch đảo ma trận trong phương pháp bình phương tối tiểu, lúc này sẽ có một phương pháp khác tối ưu hơn là Gradient Descent và các biến thể của Gradient Descent.

3. Đánh giá mức độ hoàn thành đồ án.

STT	Yêu cầu	Chưa hoàn thành	Đánh giá
1	Xây dựng mô hình hồi quy tuyến tính sử dụng toàn bộ 11 đặc trưng của dữ liệu rượu.	Không	100%
2	Tìm ra được đặc trưng nào ảnh hưởng nhất đến chất lượng của rượu.	Không	100%
3	Tự xây dựng một mô hình dự đoán chất lượng rượu của riêng bạn.	Không	100%
Tổng kết	Hoàn thành tất cả yêu cầu của đồ án.	Không	100%

V. Tài liệu tham khảo.

[1]: Machine Learning cơ bản blog (Hồi quy tuyến tính) – Vũ Hữu Tiệp.
https://machinelearningcoban.com/2016/12/28/linearregression/
[2]: Deep Learning cơ bản ebook (Linear Regression) – Nguyễn Thanh Tuấn.
https://drive.google.com/file/d/1lNjzISABdoc7SRq8tg-xkCRRZRABPCKi/view?fbclid=IwAR0HqulMwb3wSOU_NTJvnwKRny885gRiEy9szlBV-4zycInYeEOijjy5qcI
[3]: Deep Learning cơ bản blog – Nguyễn Thanh Tuấn.
https://nttuan8.com/category/deep-learning/
[4]: K-Folds cross validator – Scikit Learn.
https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html