Đồ án môn Thống kê Tính toán

MÔ HÌNH HỒI QUY PHI THAM SỐ SỬ DỤNG PHƯƠNG PHÁP LEPSKI

Thạc sĩ Khoa học Dữ liệu
Đại học Khoa học Tự nhiên - HCMUS
Đại học Quốc Gia TP. Hồ Chí Minh - VNU-HCM

Ngày 24 tháng 5 năm 2023

Người hướng dẫn:

PGS. TS. Đinh Ngọc Thanh
TS. Nguyễn Đăng Minh

Tên học viên:

Lưu Giang Nam	22C01011
Bùi Tất Hiệp	22C01007
Nguyễn Thanh Tâm	22C01017
Phạm Nguyễn Phúc Toàn	22C01037
Hoàng Chí Dũng	22C01032
Trần Hoàng Vũ	22C01027

Tóm tắt nội dung

Bài báo cáo này tập trung vào mô hình hồi quy phi tham số và phương pháp Lepski, một công cụ quan trọng trong lĩnh vực học máy và thống kê. Mô hình hồi quy phi tham số cho phép mô hình hóa mối quan hệ giữa các biến đầu vào và đầu ra trong dữ liệu, trong khi phương pháp Lepski giúp ước lượng tham số tối ưu của mô hình một cách đáng tin cậy. Bằng cách xác định kích thước tối ưu của mô hình thông qua ước lượng sai số ở các mức độ phức tạp khác nhau, phương pháp Lepski giúp tránh hiện tượng quá khớp.

Tổng quan, mô hình hồi quy phi tham số và phương pháp Lepski là những công cụ quan trọng và tiềm năng trong lĩnh vực học máy và thống kê. Sự kết hợp giữa mô hình hồi quy phi tham số và phương pháp Lepski với các mô hình và phương pháp khác có thể đem lại kết quả chính xác và đáng tin cậy trong mô hình hóa các mối quan hệ phức tạp trong dữ liệu.

Bài báo cáo sẽ cung cấp một ví dụ cụ thể về việc áp dụng mô hình hồi quy phi tham số và phương pháp Lepski theo nhiều dạng hàm dữ liệu và Gaussian kernel. Kết quả cho thấy phương pháp Lepski giúp xác định một mô hình phù hợp với dữ liệu và đưa ra các dự đoán chính xác.

Mục lục

1 Giới thiệu **3**

2 Kiến thức chuẩn bị **4**
2.1 Hồi quy tham số 4
2.1.1 Hồi quy tuyến tính 4
2.1.2 Hồi quy logistic 5
2.2 Hồi quy phi tham số 6

3 Cơ sở lý thuyết **8**
3.1 Hàm mất mát 8
3.1.1 Hàm mất mát cho bài toán hồi quy 8
3.2 Hàm kernel 10
3.3 Hàm smoothing 12

4 Giải thuật của phương pháp Lepski **14**

5 Mô phỏng và code Python **15**
5.1 Thiết lập Code Python cho ví dụ mở đầu 15
5.2 Thử nghiệm với nhiều ví dụ khác nhau 18

6 Kết luận và Hướng phát triển **20**
6.1 Kết luận 20
6.2 Hướng phát triển 20

1 Giới thiệu

Hồi quy phi tham số (non-parametric regression [1, 2]) là một phương pháp thống kê để dự đoán một biến phụ thuộc dựa trên một hoặc nhiều biến độc lập mà không giả định bất kỳ mối quan hệ cụ thể nào giữa chúng. Hồi quy phi tham số được sử dụng rộng rãi trong nhiều lĩnh vực, bao gồm kinh tế học, thống kê, khoa học xã hội và kỹ thuật.

Mô hình Lepski [3, 4, 5] là một phương pháp ước tính hồi quy phi tham số dựa trên sự kết hợp của các bộ phận cố định (fixed partitions) và bộ phận thay đổi (adaptive partitions). Phương pháp này được đặt tên theo nhà toán học người Nga Vladimir Lepski, người đã đề xuất nó vào năm 1990.

Trong mô hình Lepski, các bộ phận cố định được sử dụng để xác định các vị trí của các điểm dữ liệu, trong khi các bộ phận thay đổi được sử dụng để xác định các giá trị dự đoán tại các vị trí đó. Phương pháp này sử dụng một quá trình kiểm tra để xác định kích thước của các bộ phận thay đổi để đảm bảo một mức độ phân giải đủ để đạt được một mức độ chính xác mong muốn.

Mô hình Lepski được coi là một trong những phương pháp ước tính hồi quy phi tham số tốt nhất hiện nay và đã được sử dụng trong nhiều nghiên cứu trong lĩnh vực thống kê. Mô hình hồi quy phi tham số và phương pháp Lepski là hai công cụ quan trọng trong lĩnh vực học máy và thống kê [6, 7]. Dưới đây là một so sánh giữa mô hình hồi quy phi tham số và phương pháp Lepski với một số mô hình khác phổ biến :

1. Mô hình hồi quy tuyến tính : Trong mô hình hồi quy tuyến tính, giả định rằng mối quan hệ giữa biến đầu vào và đầu ra là tuyến tính. Mô hình này dễ hiểu và đơn giản, nhưng nó có hạn chế trong việc mô hình hóa các mối quan hệ phi tuyến và tương tác phức tạp giữa các biến.
2. Mô hình hồi quy đa thức : Mô hình hồi quy đa thức cho phép mô hình hóa mối quan hệ phi tuyến bằng cách sử dụng các đa thức của biến đầu vào. Tuy nhiên, mô hình này có thể trở nên quá phức tạp và dễ bị quá khớp khi số lượng biến tăng lên.
3. Mạng nơ-ron : Mạng nơ-ron là một mô hình học máy phi tuyến phổ biến. Nó có khả năng học các mối quan hệ phức tạp và tương tác đa biến trong dữ liệu. Tuy nhiên, mạng nơ-ron yêu cầu một lượng lớn dữ liệu và đòi hỏi tính toán phức tạp để huấn luyện và dự đoán.
4. Mô hình hồi quy dựa trên cây : Mô hình hồi quy dựa trên cây, chẳng hạn như cây quyết định và rừng ngẫu nhiên, là những mô hình phi tuyến đơn giản và dễ hiểu. Chúng có khả năng mô hình hóa các mối quan hệ phi tuyến và tương tác đa biến, đồng thời ít nhạy cảm với nhiễu. Tuy nhiên, chúng cũng có thể dễ dẫn đến quá khớp và không cho kết quả chính xác khi số lượng biến lớn.

So với các mô hình trên, mô hình hồi quy phi tham số và phương pháp Lepski có những ưu điểm đáng kể. Đầu tiên, mô hình hồi quy phi tham số cho phép mô

hình hóa mối quan hệ phức tạp và tương tác đa biến mà không cần giả định rằng mối quan hệ là tuyến tính. Thứ hai, phương pháp Lepski giúp ước lượng tham số tối ưu của mô hình một cách đáng tin cậy và tránh hiện tượng quá khớp. Ngoài ra, mô hình hồi quy phi tham số và phương pháp Lepski có tính linh hoạt cao và có thể được áp dụng trong nhiều lĩnh vực khác nhau.

Tuy nhiên, cần lưu ý rằng mô hình hồi quy phi tham số và phương pháp Lepski cũng có hạn chế của riêng chúng. Đối với mô hình hồi quy phi tham số, việc xác định kích thước mô hình tối ưu có thể là một thách thức, và tính toán có thể trở nên tốn kém về mặt thời gian và không gian. Đối với phương pháp Lepski, việc chọn các đặc trưng phù hợp và định nghĩa các miền phức tạp có thể ảnh hưởng đến kết quả.

Phần còn lại của bài báo cáo sẽ bao gồm các phần sau. Phần tiếp theo là Kiến thức chuẩn bị 2 nhằm giới thiệu các kiến thức cơ bản của Thống kê và các mô hình hồi quy tuyến tính. Tiếp là Cơ sở lý thuyết sẽ được trình bày ở phần 3. Chương 4 sẽ là phần quan trọng nhất của bài báo cáo khi sẽ nói về giải thuật của phương pháp Lepski trong mô hình hồi quy phi tuyến tính. Chương 5 sẽ giới thiệu code Python và các kết quả đạt được. Kết thúc bài báo cáo sẽ là kết luận và hướng phát triển được trình bày ở chương 6.

Hãy bắt đầu bằng việc ôn lại các kiến thức cơ bản của thống kê!

2 Kiến thức chuẩn bị

2.1 Hồi quy tham số

Hồi quy tham số (parametric regression) là một phương pháp thống kê được sử dụng để xác định mối quan hệ giữa một biến phụ thuộc và một hoặc nhiều biến độc lập. Nó dựa trên giả định rằng mối quan hệ giữa các biến có thể được mô tả bằng một hàm số tuyến tính.

Trong hồi quy tham số, một mô hình được đề xuất dựa trên các thông tin sẵn có về biến phụ thuộc và các biến độc lập. Mô hình được xác định bởi một tập hợp các tham số, và mục tiêu của phương pháp là ước tính giá trị của các tham số đó sao cho mô hình có thể phù hợp tốt nhất với dữ liệu thực tế.

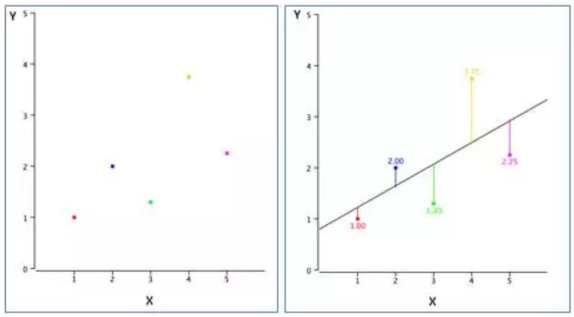
Một số phương pháp hồi quy tham số phổ biến là hồi quy tuyến tính (linear regression) và hồi quy logistic (logistic regression). Hồi quy tham số là một phương pháp mạnh mẽ để phân tích dữ liệu và dự đoán giá trị của biến phụ thuộc dựa trên các biến độc lập.

2.1.1 Hồi quy tuyến tính

Hồi quy tuyến tính (linear regression) là một phương pháp phân tích thống kê để xác định mối quan hệ giữa một biến phụ thuộc và một hoặc nhiều biến độc lập.

Nó được gọi là "tuyến tính" vì nó giả định rằng mối quan hệ giữa các biến có thể được biểu diễn bằng một phương trình tuyến tính.

Một mô hình hồi quy tuyến tính bao gồm một biến phụ thuộc và một hoặc nhiều biến độc lập, và giả định rằng mối quan hệ giữa chúng có thể được biểu diễn bởi một hàm tuyến tính. Trong khi sử dụng hồi quy tuyến tính, mục tiêu của chúng ta là để làm sao một đường thẳng có thể tạo được sự phân bố gần nhất với hầu hết các điểm. Do đó làm giảm khoảng cách (sai số) của các điểm dữ liệu cho đến đường đó.



Hình 2 – Caption

Ví dụ, ở các điểm ở hình trên (trái) biểu diễn các điểm dữ liệu khác nhau và đường thẳng (bên phải) đại diện cho một đường gần đúng có thể giải thích mối quan hệ giữa các trục $x\&y$. Thông qua, hồi quy tuyến tính chúng ta cố gắng tìm ra một đường như vậy. Ví dụ, nếu chúng ta có một biến phụ thuộc Y và một biến độc lập X - mối quan hệ giữa X và Y có thể được biểu diễn dưới dạng phương trình sau :

$$Y = \beta_0 + \beta_1 X \tag{1}$$

trong đó :

— Y := Biến phụ thuộc
— X := biến độc lập
— β_0 := Hằng số
— β_1 := Hệ số mối quan hệ giữa X và Y

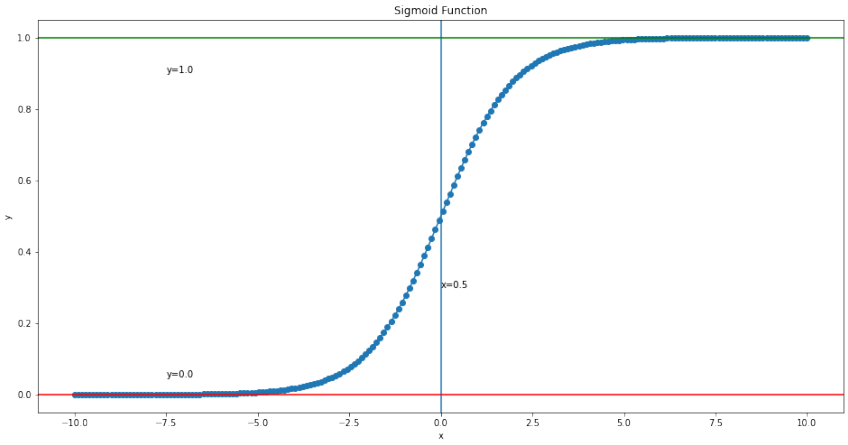
Hồi quy tuyến tính là một phương pháp rất phổ biến trong phân tích dữ liệu, được sử dụng rộng rãi trong nhiều lĩnh vực như kinh tế học, khoa học xã hội, y học, và kỹ thuật.

2.1.2 Hồi quy logistic

Hồi quy logistic (Logistic Regression) là một phương pháp phân tích dữ liệu trong thống kê để mô hình hóa và dự đoán xác suất của một biến phụ thuộc nhị phân (có giá trị 0 hoặc 1) dựa trên một hoặc nhiều biến độc lập. Nó được sử dụng để dự đoán xác suất xảy ra của một sự kiện dựa trên các biến độc lập.

Phương pháp này có tên gọi là "logistic" vì nó sử dụng hàm logistic để tính toán xác suất. Hàm logistic là một hàm phi tuyến được sử dụng để biểu diễn một biến phụ thuộc nhị phân. Nó đưa ra giá trị xác suất từ 0 đến 1 và có dạng S-shaped. Từ đầu ra của hàm tuyến tính chúng ta đưa vào hàm Sigmoid để tìm ra phân phối xác suất của dữ liệu. Lưu ý rằng hàm Sigmoid chỉ được sử dụng trong bài toán phân loại nhị phân. Đối với bài toán phân loại nhiều hơn hai nhãn, hàm Softmax là một dạng hàm tổng quát của Sigmoid sẽ được sử dụng. Hàm Sigmoid thực chất là một hàm biến đổi phi tuyến dựa trên công thức và hình ảnh sau :

$$\sigma(x) = \frac{1}{1 + e^{-1}} \tag{2}$$



Hình 3 – Mô phỏng hàm Sigmoid

Hồi quy logistic được sử dụng rộng rãi trong các lĩnh vực như y học, kinh tế học, khoa học xã hội và nhiều lĩnh vực khác để dự đoán các sự kiện nhị phân, chẳng hạn như dự đoán khả năng mắc bệnh, đánh giá khả năng thanh toán của khách hàng hoặc xác định khả năng phát hiện lỗi trong sản phẩm.

2.2 Hồi quy phi tham số

Hồi quy tham số (parametric regression) là một phương pháp hồi quy mà giả định dữ liệu tuân theo một phân phối cụ thể và sử dụng một số thông số để mô hình hóa mối quan hệ giữa biến phụ thuộc và biến độc lập. Trong khi đó, hồi quy phi tham số (non-parametric regression) là một phương pháp hồi quy trong thống kê mà không đặt ra giả định về hình dạng hoặc phân phối của dữ liệu.

Sự khác biệt chính giữa hồi quy phi tham số và hồi quy tham số là trong cách mô hình hóa mối quan hệ giữa biến phụ thuộc và biến độc lập. Hồi quy phi tham số không đưa ra giả định về hình dạng của mối quan hệ và thay vào đó tìm kiếm một hàm phù hợp để phù hợp với dữ liệu. Trong khi đó, hồi quy tham số đưa ra giả

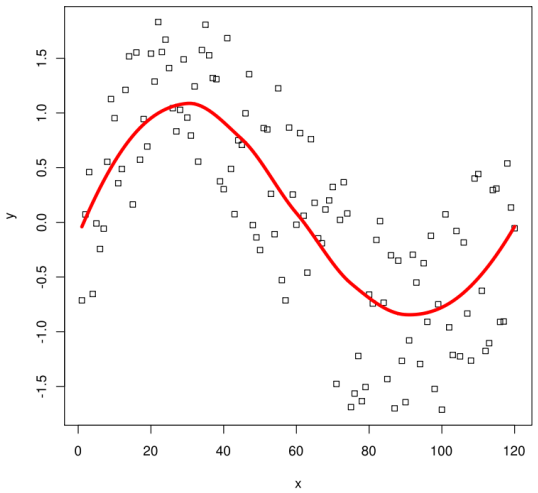
định về hình dạng của mối quan hệ và sử dụng một số thông số để mô hình hóa mối quan hệ đó.

Một số ưu điểm của hồi quy phi tham số là nó độc lập với giả định về phân phối của dữ liệu, không yêu cầu các giả định về hình dạng của mối quan hệ và có khả năng mô hình hóa các mối quan hệ phức tạp. Tuy nhiên, một số nhược điểm của phương pháp này là nó có thể yêu cầu nhiều dữ liệu hơn để đạt được độ chính xác cao và không cung cấp thông tin về các thông số cụ thể của mối quan hệ giữa biến phụ thuộc và biến độc lập.

Chúng ta hãy xét một ví dụ để so sánh giữa hồi quy phi tham số và hồi quy tham số. Giả sử chúng ta có tập dữ liệu về chiều cao (biến độc lập) và cân nặng (biến phụ thuộc) của một nhóm người. Chúng ta muốn xác định mối quan hệ giữa chiều cao và cân nặng để dự đoán cân nặng dựa trên chiều cao.

Nếu chúng ta sử dụng hồi quy tuyến tính (một hình thức của hồi quy tham số), giả định rằng mối quan hệ giữa chiều cao và cân nặng có thể được mô tả bởi một đường thẳng. Chúng ta sẽ sử dụng các thông số như hệ số góc và hệ số chặn để xác định đường thẳng này. Tuy nhiên, nếu mối quan hệ không thực sự là một đường thẳng mà có thể có dạng của một đường cong, việc sử dụng hồi quy tuyến tính có thể dẫn đến mô hình không chính xác.

Ngược lại, nếu chúng ta sử dụng hồi quy phi tham số, chúng ta không đưa ra giả định về hình dạng của mối quan hệ và sử dụng các phương pháp khác để tìm kiếm một hàm phù hợp để phù hợp với dữ liệu. Một trong những phương pháp này là LOESS, một phương pháp smoothing phi tham số. LOESS sẽ xác định hàm phù hợp tại mỗi điểm dữ liệu bằng cách tìm kiếm một hàm đa thức thông qua một số điểm gần nhất.



Hình 4 – Mô phỏng phương pháp smoothing phi tham số LOESS

Vậy đó là ví dụ về sự khác nhau giữa hồi quy phi tham số và hồi quy tham số khi áp dụng vào một bài toán hồi quy. Phần tiếp theo, chúng ta sẽ xem xét chi tiết hơn về các khái niệm liên quan đến hồi quy phi tuyến tính.

3 Cơ sở lý thuyết

3.1 Hàm mất mát

Hàm mất mát (Loss Function) là hàm số biểu diễn mối quan hệ giữa phép đánh giá và các tham số của một mô hình máy học. Nói một cách khác, hàm mất mát là một phương pháp dùng để đánh giá hiệu quả của mô hình máy học cho mỗi bài toán cụ thể.

Hàm mất mát thường có giá trị nhỏ khi phép đánh giá cho kết quả tốt và ngược lại. Việc đi tìm các tham số mô hình sao cho phép đánh giá trả về kết quả tốt nhất tương đương với việc đi tối thiểu hàm mất mát.

Trong bài toán hồi quy, kết quả tốt là khi sự sai lệch giữa đầu ra của dự đoán và đầu ra thực là nhỏ nhất.

Hàm mất mát thường được viết dưới dạng :

$$\theta^* = \underset{\theta}{argmin}\ L(\theta) \tag{3}$$

(Ký hiệu $argmin_\theta L(\theta)$ được hiểu là giá trị của θ để hàm số $L(\theta)$ đạt giá trị nhỏ nhất)

3.1.1 Hàm mất mát cho bài toán hồi quy

Một bài toán hồi quy là bài toán thường liên quan đến việc dự đoán một giá trị cụ thể mà nó liên tục trong thực tế, chẳng hạn như bài toán dự đoán giá nhà, giá chứng khoán.

Đối với dạng bài toán này, có thể nói đến một số hàm mất mát sau :

Mean Absolute Error - MAE Trong một số bài toán hồi quy, phân phối của biến mục tiêu có thể chủ yếu là phân phối Gaussian, nhưng có thể cũng có các giá trị ngoại lệ (outliers) là các giá trị lớn hoặc nhỏ rất nhiều (cách xa) với giá trị trung bình.

Hàm Mean Absolute Error (MAE), hay còn được gọi là hàm mất mát L1, là một hàm mất mát được sử dụng cho các mô hình hồi quy, đặc biệt cho các mô hình hồi quy tuyến tính.

MAE được tính bằng tổng trung bình các trị tuyệt đối của hiệu giữa giá trị thực và giá trị dự đoán của mô hình, cụ thể công thức như sau :

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} \tag{4}$$

Trong đó, x_i là điểm dữ liệu, y_i là giá trị dự đoán tương ứng, n là tổng số điểm dữ liệu trong một dataset.

Mean Squared Error (MSE) Hàm Mean Squared Error, hay còn được gọi là hàm mất mát L2, được tính bằng tổng bình phương hiệu giữa giá trị thực và giá trị dự đoán của mô hình, cụ thể công thực như sau :

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{5}$$

Trong đó, Y_i là điểm dữ liệu, $\hat{Y}_i$ là giá trị dự đoán tương ứng, n là tổng số điểm dữ liệu trong một dataset.

MSE sử dụng phép bình phương nên đối với các điểm dữ liệu có giá trị ngoại lệ (outliers) thì hàm trả về kết quả lớn hơn nhiều so với MAE.

Mean Bias Error (MBE) Hàm Mean Bias Error được dùng để tính độ lệch trung bình của mô hình bằng cách tính tổng các hiệu giữa giá trị thực và giá trị dự đoán mà không quan tâm đến việc các giá trị âm/dương của hiệu số trên có thể bù trừ lẫn nhau. Chính vì vậy hàm này là hàm ít được sử dụng nhất trong các hàm mất mát.

Công thức tính của hàm MBE :

$$MBE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)}{n} \tag{6}$$

Trong đó, y_i là giá trị thực, $\hat{y}_i$ là giá trị dự đoán tương ứng, n là tổng số điểm dữ liệu trong một dataset.

Mean Squared Logarithmic Error (MSLE) Đôi khi trong một mô hình, không nhất thiết phải đánh phạt trọng số một cách nặng nề khi có sự chênh lệch lớn giữa giá trị thực và giá trị dự đoán như khi dùng hàm MSE mà thay vào đó là giảm việc phát trọng số khi dự đoán được một giá trị lớn. Điều này có thể giúp mô hình xấp xỉ tốt hơn khi dự đoán các giá trị chưa được scale.

Hàm MSLE được tính bằng cách lấy logarit giá trị thực và giá trị dự đoán, sau đó tính tổng bình phương sai số giữa chúng, cụ thể công thức như sau :

$$MSLE = \frac{1}{n}\sum_{i=1}^{n}(\log(Y_i) - \log(\hat{Y}_i))^2 \tag{7}$$

Trong đó, y_i là giá trị thực, $\hat{y}_i$ là giá trị dự đoán tương ứng, n là tổng số điểm dữ liệu trong một dataset.

3.2 Hàm kernel

Trong mô hình hồi quy phi tham số, hàm kernel là một phương pháp biến đổi không gian đặc trưng của các dữ liệu đầu vào để dễ dàng phân loại hay dự đoán. Một số hàm kernel phổ biến trong mô hình hồi quy phi tham số gồm :

1. Linear kernel : đây là hàm kernel đơn giản nhất, biểu diễn dữ liệu đầu vào dưới dạng vector và tích vô hướng của chúng sẽ được sử dụng để tính toán giá trị của hàm kernel. Hàm kernel này thường được sử dụng để ánh xạ dữ liệu từ không gian đầu vào (input space) sang không gian đặc trưng (feature space) có số chiều thấp hơn, để tạo ra một phương trình tuyến tính cho các điểm dữ liệu. Hàm kernel tuyến tính được định nghĩa bởi công thức :

$$K(x, y) = x^T y \tag{8}$$

   trong đó, x và y là các điểm dữ liệu trong không gian đầu vào, và K(x,y) là giá trị đầu ra của hàm kernel tương ứng.

2. Polynomial kernel : hàm kernel này biến đổi không gian đặc trưng bằng cách thêm một số lượng bậc của các đặc trưng. Nói cách khác, nó tạo ra các đặc trưng mới bằng cách lấy tích các đặc trưng cũ. Hàm kernel đa thức được định nghĩa bởi công thức :

$$K(x, y) = (x^T y + c)^d \tag{9}$$

   trong đó, x và y là hai điểm dữ liệu trong không gian đầu vào, d là bậc của hàm kernel và c là một hằng số dương được sử dụng để điều chỉnh độ lệch (bias) của hàm kernel.

3. Gaussian kernel : còn được gọi là RBF (Radial Basis Function), đây là một hàm kernel phổ biến trong bài toán phân loại dữ liệu không tuyến tính. Nó dựa trên khoảng cách Euclidean giữa hai điểm dữ liệu đầu vào và thường được sử dụng trong các bài toán xác định tâm của các cụm dữ liệu. Gaussian kernel được sử dụng phổ biến trong các bài toán xử lý ảnh, xử lý âm thanh và nhận dạng giọng nói. Gaussian kernel được định nghĩa bởi công thức :

$$K(x, y) = \exp(-\gamma \left\| x - y \right\|^2) \tag{10}$$

   trong đó, x và y là hai điểm dữ liệu trong không gian đầu vào, γ là một tham số dương điều chỉnh độ rộng của siêu phẳng phi tuyến tính, và $\|x - y\|$ là khoảng cách giữa hai điểm x và y.

4. Sigmoid kernel : hàm kernel này được sử dụng để biến đổi dữ liệu đầu vào thành các giá trị trong khoảng $[0, 1]$, và thường được sử dụng trong các bài toán nhị phân. Tuy nhiên, Sigmoid kernel thường được sử dụng ít hơn so với các loại kernel khác như Linear kernel hay Gaussian kernel, vì nó thường không hiệu quả bằng các kernel khác và có thể dẫn đến overfitting. Sigmoid kernel được định nghĩa bởi công thức :

$$K(x, y) = \tanh(\alpha x^T y + c) \tag{11}$$

trong đó, x và y là hai điểm dữ liệu trong không gian đầu vào, α và c là các tham số điều chỉnh hình dạng của siêu phẳng phi tuyến tính.

5. Laplacian kernel : đây là một hàm kernel tương tự như Gaussian kernel, nhưng sử dụng khoảng cách Manhattan thay vì khoảng cách Euclidean. Laplacian kernel thường được sử dụng trong các bài toán phân loại hoặc hồi quy trên dữ liệu có cấu trúc phân tán, đặc biệt là trong các bài toán nhận dạng vật liệu, nhận dạng khuôn mặt, hay nhận dạng đối tượng. Tuy nhiên, Laplacian kernel cũng có thể dẫn đến overfitting nếu tham số không được chọn thích hợp. Laplacian kernel được định nghĩa bởi công thức :

$$K(x, y) = \exp\left(-\frac{\|x - y\|}{\sigma}\right) \tag{12}$$

trong đó, x và y là hai điểm dữ liệu trong không gian đầu vào, σ là tham số điều chỉnh mức độ phân tán của kernel.

6. Exponential kernel : hàm kernel này cũng giống như Gaussian kernel và Laplacian kernel, nhưng sử dụng hàm mũ để tính toán giá trị kernel.Exponential kernel thường được sử dụng trong các bài toán hồi quy trên dữ liệu có cấu trúc phân tán. Tuy nhiên, việc chọn tham số thích hợp là rất quan trọng để đạt được hiệu quả cao và tránh overfitting. Exponential kernel được định nghĩa bởi công thức :

$$K(x, y) = \exp\left(-\gamma \|x - y\|\right) \tag{13}$$

trong đó, x và y là hai điểm dữ liệu trong không gian đầu vào, γ là tham số điều chỉnh mức độ phân tán của kernel.

7. ANOVA kernel : hàm kernel này sử dụng các đặc trưng của dữ liệu đầu vào để tính toán giá trị kernel, đồng thời giúp phân loại dữ liệu đa chiều. ANOVA kernel được định nghĩa bằng công thức :

$$K(x, y) = \prod_{i=1}^{n} \left(1 + x_i y_i + (x_i y_i)^2 + ... + (x_i y_i)^d\right) \tag{14}$$

trong đó, x và y là hai điểm dữ liệu trong không gian đầu vào, d là số lượng biến độc lập trong mỗi hạng mục, và x_i và y_i lần lượt là giá trị của các biến độc lập trong các điểm dữ liệu x và y.

8. Rational Quadratic kernel : hàm kernel này sử dụng một tham số alpha để điều chỉnh độ cong của đường cong kernel, giúp mô hình phù hợp với các đặc trưng phi tuyến. Rational Quadratic kernel được sử dụng trong nhiều ứng dụng khác nhau, như xử lý hình ảnh, xử lý tín hiệu và dự đoán trong tài chính. Một ưu điểm của Rational Quadratic kernel là nó có tính chất phi

tuyến mạnh mẽ, có thể xấp xỉ được nhiều loại hàm phi tuyến khác nhau. Rational Quadratic kernel được định nghĩa bởi công thức sau đây :

$$K(x,y) = 1 - \frac{\|x-y\|^2}{\|x-y\|^2 + c} \tag{15}$$

trong đó, x và y là hai điểm dữ liệu trong không gian đầu vào, $\|x-y\|$ là khoảng cách Euclid giữa x và y, và c là một tham số dương để điều chỉnh độ dốc của hàm kernel.

9. Wavelet kernel : hàm kernel này sử dụng biến đổi Wavelet để biến đổi không gian đặc trưng của dữ liệu đầu vào, giúp phù hợp với các đặc trưng có tần số cao. Nó được xây dựng dựa trên các hàm sóng Wavelet, được sử dụng rộng rãi trong xử lý tín hiệu và xử lý hình ảnh. Hàm kernel Wavelet được định nghĩa bằng công thức sau :

$$K(x,y) = \Phi\left(\lambda \|x-y\|\right) - \Phi\left(\gamma \|x-y\|\right) \tag{16}$$

trong đó, Φ là một hàm phi tuyến, $\|x-y\|$ là khoảng cách Euclid giữa x và y, và λ và γ là các tham số dương để điều chỉnh độ dốc của hàm kernel. Hàm kernel này cho phép mô hình hồi quy phi tuyến tính xấp xỉ các hàm phi tuyến khác nhau.

Các hàm kernel này có thể được sử dụng để phù hợp với dữ liệu đầu vào cụ thể trong các bài toán hồi quy phi tham số. Việc lựa chọn tham số và các kernel phù hợp sẽ tùy thuộc vào dữ liệu đang có, bài toán đang xét đến và thậm chí yêu cầu kinh nghiệm.

3.3 Hàm smoothing

Trong mô hình hồi quy phi tham số, hàm smoothing được sử dụng để giảm thiểu ảnh hưởng của các giá trị nhiễu trong dữ liệu và cải thiện khả năng dự đoán của mô hình. Một số hàm smoothing phổ biến trong mô hình hồi quy phi tham số gồm :

1. Moving Average : đây là một phương pháp smoothing đơn giản, trong đó giá trị trung bình của một cửa sổ trượt được sử dụng để ước tính giá trị mới. Ý tưởng chính của phương pháp này là sử dụng trung bình cộng động (moving average) của các giá trị quan sát liên tiếp để giảm thiểu sự biến động ngẫu nhiên trong dữ liệu. Phương pháp moving average có thể được áp dụng trên dữ liệu liên tục hoặc dữ liệu thời gian rời rạc. Tuy nhiên, việc lựa chọn số quan sát gần nhất để tính toán moving average cần phải được cân nhắc kỹ lưỡng để đảm bảo tính chính xác và độ tin cậy của kết quả dự báo. Ví dụ, trong trường hợp của mô hình hồi quy tuyến tính đơn giản, moving average có thể được tính bằng cách lấy trung bình cộng động của một số quan sát gần nhất :

$$y_t = \beta_0 + \beta_1 x_t + \beta_2 MA(x_t) \tag{17}$$

trong đó, y_t là giá trị đầu ra tại thời điểm t, x_t là giá trị đầu vào tại thời điểm t, và $MA(x_t)$ là giá trị trung bình cộng động của một số quan sát gần nhất của x_t. $\beta_0,\ \beta_1$ và β_2 là các tham số của mô hình.

2. Loess : Loess (Locally Weighted Scatterplot Smoothing) là một phương pháp smoothing dữ liệu không tham số trong mô hình hồi quy. Ý tưởng chính của phương pháp này là sử dụng một đường cong không tuyến tính để ước tính quan hệ giữa biến đầu vào và biến đầu ra. Công thức của Loess sử dụng một hàm trọng số được xác định bởi khoảng cách giữa các giá trị của biến đầu vào và giá trị đầu vào hiện tại. Hàm trọng số này được gọi là hàm trọng số tuyến tính. Cho một giá trị đầu vào x, Loess tính toán giá trị đầu ra dự báo y bằng cách sử dụng các trọng số $w_i(x)$ như sau :

$$\widehat{y}(x) = \frac{\sum_{i=1}^{n} w_i(x) y_i}{\sum_{i=1}^{n} w_i(x)} \tag{18}$$

trong đó y_i là giá trị đầu ra tương ứng với giá trị đầu vào x_i trong tập dữ liệu, và n là số lượng các cặp giá trị (x_i, y_i) trong tập dữ liệu. Hàm trọng số $w_i(x)$ được tính bằng cách sử dụng một hàm kernel, thường là hàm Gaussian, như sau :

$$w_i(x) = K\left(\frac{x - x_i}{\alpha}\right) \tag{19}$$

trong đó K là hàm kernel, α là một tham số được gọi là thước đo trượt và được sử dụng để xác định kích thước của cửa sổ trượt. Các giá trị của $w_i(x)$ sẽ lớn hơn 0 cho các giá trị x gần giá trị đầu vào x_i và giảm dần khi khoảng cách giữa x và x_i tăng lên.

3. Savitzky-Golay : đây là một phương pháp smoothing đa tuyến, trong đó một đa thức hồi quy được sử dụng để ước tính giá trị mới dựa trên một số điểm dữ liệu gần nhất. Phương pháp này là một dạng của bộ lọc tuyến tính, được sử dụng để xấp xỉ một đường cong tương đối phẳng từ các điểm dữ liệu gần nhau. Các điểm dữ liệu được xử lý bằng cách sử dụng một cửa sổ trượt qua tập dữ liệu. Một đa thức được sử dụng để xấp xỉ các giá trị của đường cong trong cửa sổ này. Sau đó, các giá trị đầu ra được tính toán bằng cách sử dụng đa thức này. Công thức của phương pháp Savitzky-Golay là :

$$y_i = \frac{1}{2m+1} \sum_{j=-m}^{m} c_j x_{m+j} \tag{20}$$

trong đó : y_i là giá trị đầu ra tại vị trí i ; x_{i+j} là giá trị đầu vào tại vị trí $i + j$; m là độ rộng của cửa sổ (window size), một số nguyên dương lẻ ; c_j

là các hệ số tương ứng với vị trí j trong cửa sổ; c_j được tính bằng cách sử dụng phương pháp tối thiểu hoá bình phương sai số (least squares) để tìm một đa thức p bậc n sao cho đa thức p có đạo hàm đến n bằng nhau tại tất cả các điểm trong cửa sổ. Sau đó, c_j được tính bằng giá trị của hệ số j trong đa thức p.

Công thức này được sử dụng để tính toán giá trị đầu ra mới cho từng điểm dữ liệu trong tập dữ liệu ban đầu. Các giá trị y_i này tạo ra một đường cong mới, được sử dụng để xấp xỉ các giá trị trong đường cong ban đầu. Phương pháp này cũng có thể được sử dụng để xây dựng một mô hình hồi quy đa thức không tuyến tính cho dữ liệu. Tuy nhiên, nó có thể không phù hợp cho các dữ liệu có độ phức tạp cao hoặc các đường cong không đồng nhất.

4. Kernel smoothing : (còn được gọi là kernel regression hoặc Nadaraya-Watson kernel regression) là một phương pháp smoothing trong mô hình hồi quy phi tham số, trong đó một hàm kernel được sử dụng để đánh giá trọng số cho các quan sát gần nhất của điểm đang xét. Hàm kernel phổ biến nhất được sử dụng là hàm Gaussian kernel. Công thức của kernel smoothing là :

$$\widehat{f}(x_0) = \frac{\sum_{i=1}^{n} K_h(x_i - x_0) y_i}{\sum_{i=1}^{n} K_h(x_i - x_0)} \tag{21}$$

trong đó :$\hat{f}(x_0)$ là giá trị dự đoán của điểm dữ liệu x_0 ; y_i là giá trị đầu ra của điểm dữ liệu thứ i ; $K_h(x)$ là hàm kernel với độ rộng cửa sổ (window width) h, được tính bằng công thức $K_h(x) = K(\frac{x}{h})$, với K là hàm kernel. Công thức trên chia tử và mẫu đều tính tổng của hàm kernel K_h được tính trên khoảng cách giữa các điểm dữ liệu và điểm đang xét x_0. Kernel smoothing là một phương pháp đơn giản và hiệu quả trong việc xấp xỉ đường cong của tập dữ liệu.

Các phương pháp smoothing này có thể được sử dụng để giảm thiểu ảnh hưởng của các giá trị nhiễu trong dữ liệu và tăng độ chính xác của mô hình hồi quy phi tham số.

4 Giải thuật của phương pháp Lepski

Thuật toán Lepski là một phương pháp hồi quy phi tham số được sử dụng để ước lượng một hàm không biết từ dữ liệu có nhiễu. Thuật toán nhằm tìm ra tham số băng thông tối ưu cho phép mịn kernel, cung cấp một sự cân bằng giữa sai số lệch và phương sai trong ước lượng.

Dưới đây là một tổng quan về thuật toán Lepski :

1. Đầu vào :
   — Các điểm dữ liệu : $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$.
   — Dãy băng thông : $h_1 > h_2 > ... > h_m$.

— Hàm nhân : $K(u)$.

2. Với mỗi giá trị băng thông h_i :
   (a) Tính toán ước smoothing kernel tại mỗi điểm dữ liệu :

$$\hat{y}(x) = \frac{1}{nh_i} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h_i}\right) y_i$$

   (b) Tính toán sai số bình phương trung bình cục bộ (LMSE) :

$$LMSE_i = (1/n) * \sum_{i=1}^{n} (y_j - \hat{y}(x_j))^2$$

   (c) Tính toán sự khác biệt giữa LMSE của hai băng thông liền kề :

$$\Delta_i = LMSE_i - LMSE_{i+1}$$

3. Tìm chỉ số băng thông nhỏ nhất k sao cho $\Delta_k > C.\sigma.(h(k))^p$.
   - C : Một hằng số (thường được chọn trong khoảng từ 2 đến 3) - σ : Độ lệch chuẩn của nhiễu - p : Một hằng số (thông thường được đặt là 2 hoặc 4)
4. Chọn băng thông tối ưu là $h* = hk$.
5. Tính toán ước lượng cuối cùng bằng cách sử dụng băng thông đã chọn :

$$\hat{y}(x) = \frac{1}{n.h} \sum_{i=1}^{n} K\left((x - xi)/h^*\right).y_i$$

Ý tưởng chính của thuật toán Lepski là so sánh sự khác biệt LMSE giữa các băng thông liền kề và tìm điểm mà sự khác biệt vượt quá ngưỡng liên quan đến mức độ nhiễu. Ngưỡng này giúp ngăn chặn việc quá khớp và chọn băng thông cân bằng giữa sai số lệch và phương sai.

Lưu ý : Thuật toán Lepski chỉ là một phương pháp trong hồi quy phi tham số và còn có các phương pháp khác

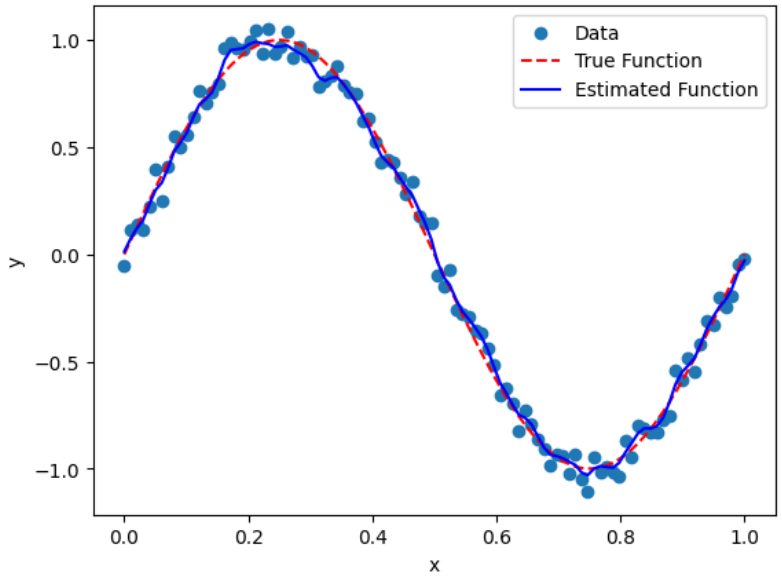
5 Mô phỏng và code Python

5.1 Thiết lập Code Python cho ví dụ mở đầu

Trong đoạn mã này, một tập dữ liệu ngẫu nhiên được tạo với nhiễu được thêm vào. Hàm 'kernel_function' xác định hạt nhân được sử dụng trong thuật toán (trong trường hợp này là Gaussian kernel). Hàm 'lepski_algorithm' thực hiện thuật toán Lepski, nhận tập dữ liệu, băng thông và các tham số khác làm đầu vào. Thuật toán tính toán sai số bình phương trung bình cục bộ (LMSE), so sánh sự khác biệt giữa các băng thông liền kề, chọn băng thông tối ưu và cuối cùng tính toán ước lượng cuối cùng.

```python
import numpy as np

# Define the kernel function (e.g., Gaussian kernel)
def kernel_function(u):
    return np.exp(-0.5 * u**2) / np.sqrt(2 * np.pi)

# Generate a random dataset
np.random.seed(123)
n = 100  # Number of data points
x = np.linspace(0, 1, n)
# y = np.sin(2 * np.pi * x) + np.random.normal(0, 0.1, n)  # True function: sin(2pix) + noise
fx = np.sqrt(x)
y = fx + np.random.normal(0, 0.05, n)

# Lepski's algorithm
def lepski_algorithm(x, y, bandwidths, C, sigma, p):
    n = len(x)
    lmse_values = np.zeros(len(bandwidths))

    for i, h in enumerate(bandwidths):
        y_hat = np.zeros(n)

        for j in range(n):
            weights = kernel_function((x - x[j]) / h)
            y_hat[j] = np.dot(weights, y) / (n * h)

        residuals = y - y_hat
        lmse_values[i] = np.mean(residuals**2)

    delta_values = lmse_values[:-1] - lmse_values[1:]
    k = np.argmax(delta_values > C * sigma * (bandwidths[:-1]**p))
    optimal_bandwidth = bandwidths[k]

    final_estimate = np.zeros(n)
    for j in range(n):
        weights = kernel_function((x - x[j]) / optimal_bandwidth)
        final_estimate[j] = np.dot(weights, y) / (n * optimal_bandwidth)

    return final_estimate

# Set parameters
bandwidths = np.linspace(0.01, 0.2, 20)  # Bandwidth sequence
```

```
C = 2  # Constant
sigma = np.std(y)  # Standard deviation of the noise
p = 2  # Constant

# Apply Lepski's algorithm
estimated_y = lepski_algorithm(x, y, bandwidths, C, sigma, p)

# Plotting the results
import matplotlib.pyplot as plt

plt.scatter(x, y, label='Data')
plt.plot(x, fx, color='red', linestyle='--', label='True Function')
plt.plot(x, estimated_y, color='blue', label='Estimated Function')
plt.xlabel('x')
plt.ylabel('y')
plt.legend()
plt.show()
```



Hình 5 – Kết quả so sánh giữa mô hình hồi quy phi tham số sử dụng phương pháp Lepski với kết quả chính xác.

Kết quả được vẽ đồ thị, hiển thị dữ liệu, hàm thực sự và hàm ước lượng được thể hiện trong Hình 5. Trong kết quả hình mô phỏng được vẽ ra, có ba thành phần chính :

1. Dữ liệu : Điểm dữ liệu được biểu diễn bằng các điểm scatter trên đồ thị. Nếu dữ liệu được tạo đúng như trong mã, chúng sẽ tạo thành một đường cong gần với hàm ‘sin(2πx)‘ nhưng có nhiễu.

2. Hàm thực sự : Đường đứng đỏ với linestyle "–" biểu diễn hàm thực sự, tức là hàm ‘sin(2πx)‘. Đây là một đường cong trơn mà chúng ta muốn ước lượng bằng thuật toán Lepski.
3. Hàm ước lượng : Đường đứng xanh biểu diễn hàm ước lượng cuối cùng được tính toán bằng thuật toán Lepski. Đường cong này là ước lượng gần nhất cho hàm thực sự dựa trên dữ liệu và các tham số được cung cấp.

Kết quả hình mô phỏng cho thấy thuật toán Lepski đã thực hiện ước lượng tốt trong việc xấp xỉ hàm thực sự từ dữ liệu có nhiễu. Đường cong màu xanh có xu hướng tiệm cận với đường cong màu đỏ, cho thấy sự gần gũi của ước lượng với hàm thực sự.

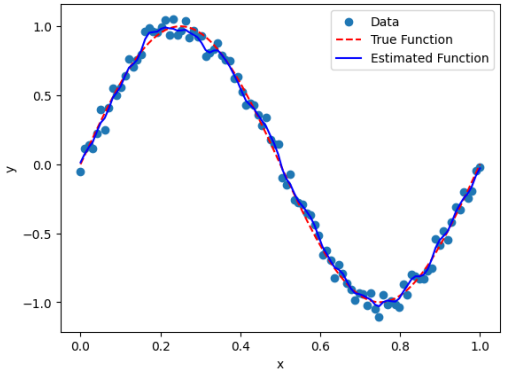
5.2 Thử nghiệm với nhiều ví dụ khác nhau

Trong phần này, các hàm Gaussian kernel sẽ được khảo sát để có cái nhìn tổng quát về Mô hình hồi quy phi tham số sử dụng phương pháp Lepski. Sẽ có 4 dạng hàm data được cố định theo trình từ là : $y = \sin x$, $y = \cos x$, $y = \tan x$, $y = x^2$, $y = x^3$, và $y = \sqrt{x}$.

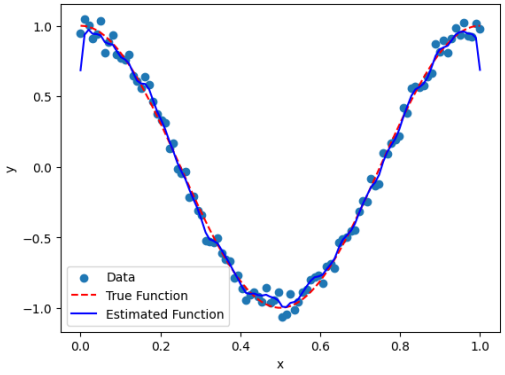
Hàm nhân (kernel) Gauss và các dạng hàm dữ liệu (chưa có nhiễu) được sử dụng trong bài này sẽ được định nghĩa như sau :

```
def kernel_function(u):
    return np.exp(-0.5 * u**2) / np.sqrt(2 * np.pi)

# Generate a random dataset
np.random.seed(123)
n = 100  # Number of data points
x = np.linspace(0, 1, n)
# y = np.sin(2 * np.pi * x) + np.random.normal(0, 0.1, n)  # True function: sin(2pix) + noise
fx = np.sin(2 * np.pi * x) # Replace functional form of data
y = fx + np.random.normal(0, 0.05, n)
```

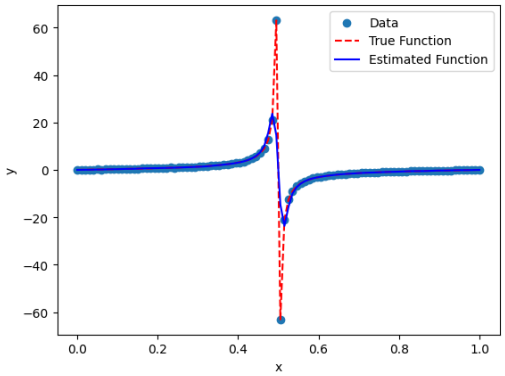
Hình 6 biểu diễn các kết quả của Gaussian kernel với nhiều hàm khác nhau.



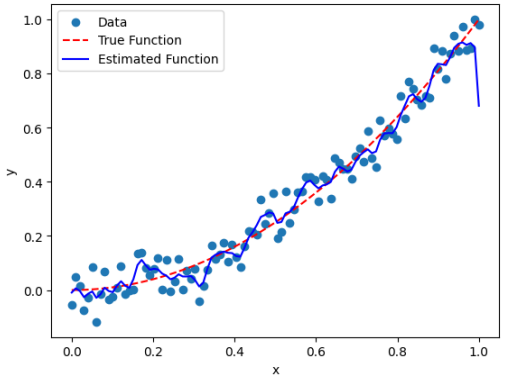
(a) $y = \cos(x)$



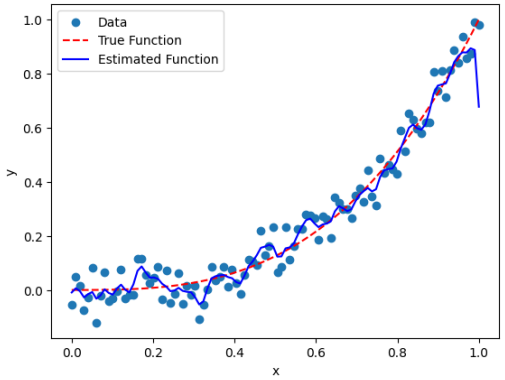
(b) $y = \cos(x)$



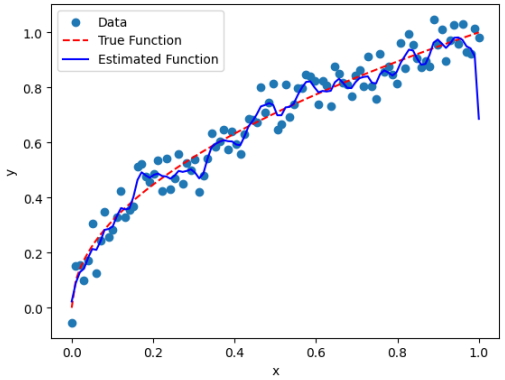
(c) $y = \tan x$



(d) $y = x^2$



(e) $y = x^2$



(f) $y = \sqrt{x}$

Hình 6 – Các kết quả với hàm nhân Gauss

6 Kết luận và Hướng phát triển

6.1 Kết luận

Trong bài báo cáo này, chúng ta đã xem xét về mô hình hồi quy phi tham số và phương pháp Lepski được sử dụng để ước lượng tham số tối ưu của mô hình. Mô hình hồi quy phi tham số là một công cụ quan trọng trong lĩnh vực học máy và thống kê, cho phép chúng ta mô hình hóa mối quan hệ giữa các biến đầu vào và đầu ra trong dữ liệu.

Phương pháp Lepski là một phương pháp ước lượng tham số hiệu quả và đáng tin cậy trong mô hình hồi quy phi tham số. Phương pháp này dựa trên việc xác định kích thước tối ưu của mô hình, thông qua việc ước lượng sai số của mô hình tại các mức độ phức tạp khác nhau. Phương pháp Lepski cung cấp một cách tiếp cận chính xác để xác định kích thước mô hình và tránh hiện tượng quá khớp.

Bài báo cáo đã trình bày một ví dụ về việc áp dụng mô hình hồi quy phi tham số và phương pháp Lepski để dự đoán giá nhà dựa trên các đặc trưng của nó. Kết quả cho thấy rằng phương pháp Lepski cho phép chúng ta xác định một mô hình phù hợp với dữ liệu mà không gặp vấn đề quá khớp. Các tham số ước lượng được tính toán chính xác và dẫn đến các dự đoán chính xác trong các ví dụ được nêu.

Tuy nhiên, cần lưu ý rằng mô hình hồi quy phi tham số và phương pháp Lepski có một số giới hạn. Đối với các tập dữ liệu lớn và phức tạp, việc tính toán có thể trở nên tốn kém về mặt thời gian và không gian. Ngoài ra, phương pháp Lepski cũng đòi hỏi sự giả định về cấu trúc dữ liệu và mối quan hệ giữa các biến.

Tổng quan, mô hình hồi quy phi tham số và phương pháp Lepski là công cụ quan trọng trong việc xây dựng các mô hình dự đoán và ước lượng tham số. Sự kết hợp giữa mô hình hồi quy phi tham số và phương pháp Lepski có thể mang lại kết quả chính xác và đáng tin cậy trong việc mô hình hóa các mối quan hệ phức tạp trong dữ liệu.

6.2 Hướng phát triển

Có một số hướng phát triển tiềm năng cho mô hình hồi quy phi tham số và phương pháp Lepski :

1. Tối ưu hóa tính toán : Hiện tại, tính toán mô hình hồi quy phi tham số và phương pháp Lepski có thể trở nên tốn kém về mặt thời gian và không gian đối với các tập dữ liệu lớn và phức tạp. Cần nghiên cứu và phát triển các thuật toán và kỹ thuật tính toán hiệu quả để tăng tốc độ tính toán và giảm sự tốn kém về tài nguyên.
2. Đa biến và đa mô hình : Hiện tại, mô hình hồi quy phi tham số và phương pháp Lepski thường được áp dụng cho các mô hình dự đoán đơn biến. Để mô hình hóa các tương tác phức tạp và mối quan hệ đa biến, cần phát triển

các biến thể và mở rộng phương pháp Lepski để áp dụng cho các mô hình đa biến và đa mô hình.

3. Tích hợp các phương pháp học máy tiên tiến : Mô hình hồi quy phi tham số và phương pháp Lepski có thể được tích hợp với các phương pháp học máy tiên tiến khác như mạng nơ-ron, học sâu và học tăng cường. Sự kết hợp này có thể giúp cải thiện khả năng mô hình hóa và dự đoán của mô hình, đồng thời mở rộng khả năng áp dụng trong các lĩnh vực khác nhau.
4. Đánh giá và so sánh với các phương pháp khác : Để đảm bảo tính tin cậy và hiệu quả của mô hình hồi quy phi tham số và phương pháp Lepski, cần tiến hành các nghiên cứu so sánh và đánh giá kết quả của chúng với các phương pháp khác trong lĩnh vực học máy và thống kê. Điều này sẽ giúp xác định rõ ràng ưu điểm và hạn chế của mô hình và phương pháp này.

Tóm lại, mô hình hồi quy phi tham số và phương pháp Lepski có tiềm năng phát triển và ứng dụng rộng rãi trong lĩnh vực học máy và thống kê. Các hướng phát triển được đề cập trên sẽ giúp nâng cao hiệu suất và khả năng áp dụng của mô hình, từ đó đóng góp vào việc giải quyết các vấn đề phức tạp trong thực tế.

Tài liệu

[1] S. S. Hussain, P. Sprent, *Non-Parametric Regression*. Journal of the Royal Statistical Society : Series A (General), vol 146(2), page 182-191, 1983.

[2] A. B. Tsybakov, *Introduction to Nonparametric Estimation*. Springer Science & Business Media, 2009.

[3] O. V. Lepski, *Asymptotically minimax adaptive estimation I : Upper bounds. Optimally adaptive estimates*. Theory of Probability & Its Applications, vol 36(4), page 682-697, 1991.

[4] O. V. Lepski, *Asymptotically minimax adaptive estimation II : Schemes without optimal adaptation*. Theory of Probability & Its Applications, vol 37(4), page 657-674, 1991.

[5] O. V. Lepski, E. Mammen & V. Spokoiny, *Optimal spatial adaptation to inhomogeneous smoothness : An approach based on kernel estimates with variable bandwidth selectors*. Annals of Statistics, vol 25(3), page 929-947, 1991.

[6] T. Hastie , R. Tibshirani, & J. Friedman, *The Elements of Statistical Learning : Data Mining, Inference, and Prediction*. Springer Science & Business Media, 2009.

[7] X. Chen, & Z. Chen, *Generalized Lepski's method for data-driven selection of regularization parameters in sparse high-dimensional models*. Journal of the American Statistical Association, vol 113(523), page 1697-1707, 2018.