

## Лабораторна 6: Нелінійні моделі.

Розглянемо дані Wage

```
> library(ISLR)
> attach(Wage)
```

### Поліноміальна регресія та східчасті функції

Пристосуємо модель використовуючи наступну команду

```
> fit=lm(wage~poly(age, 4), data=Wage)
> coef(summary(fit))
   Estimate Std. Error t value Pr(>|t|)
(Intercept) 111.704     0.729 153.28 <2e-16
poly(age, 4)1 447.068    39.915 11.20 <2e-16
poly(age, 4)2 -478.316    39.915 -11.98 <2e-16
poly(age, 4)3 125.522    39.915   3.14 0.0017
poly(age, 4)4 -77.911    39.915  -1.95 0.0510
```

Цей синтаксис оцінює лінійну модель, використовуючи функцію lm(), щоб передбачити заробітну плату з використанням полінома четвертого ступеня за age: poly(age, 4). Команда poly() дозволяє нам уникнути необхідності вписувати довгу формулу зі степенями age. Функція повертає матрицю, стовпці якої є базою з ортогональних поліномів, що по суті означає, що кожен стовпець є лінійною комбінацією змінних age, age<sup>2</sup>, age<sup>3</sup> та age<sup>4</sup>.

Однак ми також можемо використовувати poly() для отримання age, age<sup>2</sup>, age<sup>3</sup> та age<sup>4</sup> безпосередньо, якщо ми віддаємо перевагу такому підходу. Ми можемо зробити це, використовуючи аргумент raw = TRUE для функція poly(). Ми побачимо, що це не впливає на модель, але впливає на оцінки відповідних коефіцієнтів.

```
> fit2=lm(wage~poly(age, 4, raw=T), data=Wage)
> coef(summary(fit2))
   Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.84e+02  6.00e+01  -3.07 0.002180
poly(age, 4, raw = T)1 2.12e+01  5.89e+00   3.61 0.000312
poly(age, 4, raw = T)2 -5.64e-01  2.06e-01  -2.74 0.006261
poly(age, 4, raw = T)3  6.81e-03  3.07e-03   2.22 0.026398
poly(age, 4, raw = T)4 -3.20e-05  1.64e-05  -1.95 0.051039
```

Є кілька інших способів оцінювання цієї моделі, наприклад

```
> fit2a=lm(wage~age+I(age^2)+I(age^3)+I(age^4),data=Wage)
> coef(fit2a)
(Intercept)           age      I(age^2)      I(age^3)      I(age^4)
-1.84e+02       2.12e+01     -5.64e-01      6.81e-03     -3.20e-05
```

Це просто створює поліноміальні базисні функції на льоту, а для правильної інтерпретації таких речей, як, наприклад, age<sup>2</sup>, використовується функція I().

```
> fit2b=lm(wage~cbind(age,age^2,age^3,age^4),data=Wage)
```

Це робить те саме, але компактніше, використовуючи функцію cbind() для побудови матриці з колекції векторів. Тепер ми створюємо сітку значень для age, в яких ми хочемо побудувати передбачення, і потім з допомогою функції predict() будуємо їх, вказавши попередньо, що ми хочемо обчислити також стандартне відхилення.

```
> agelims=range(age)
> age.grid=seq(from=agelims[1],to=agelims[2])
> preds=predict(fit,newdata=list(age=age.grid),se=TRUE)
> se.bands=cbind(preds$fit+2*preds$se.fit,preds$fit-2*preds$se.fit)
```

Нарешті, ми побудуємо графік та додамо на нього отриману оцінку полінома 4-ого степеня.

```
> par(mfrow=c(1,2),mar=c(4.5,4.5,1,1),oma=c(0,0,4,0))
> plot(age,wage,xlim=agelims,cex=.5,col="darkgrey")
> title("Degree-4 Polynomial",outer=T)
> lines(age.grid,preds$fit,lwd=2,col="blue")
> matlines(age.grid,se.bands,lwd=1,col="blue",lty=3)
```

Тут аргументи mar та oma для par() дозволяють нам контролювати поля графіку, а функція title() створює заголовок фігури, який охоплює обидва підграфіки.

Ми згадували раніше, що вибір, чи ортогональний набір базисних функцій чи ні буде виконувати функцію poly(), не впливає на отриману модель в тому сенсі, що отримані прогнозні значення в обох випадках ідентичні

```
> preds2=predict(fit2,newdata=list(age=age.grid),se=TRUE)
> max(abs(preds$fit-preds2$fit))
[1] 7.39e-13
```

Оцінюючи поліноміальну регресію, ми повинні визначитися зі степенем полінома, який слід використовувати. Один із способів зробити це - використання тестів гіпотез. Ми оцінемо моделі від лінійної до поліноміальної

степеня 5 і визначимо найпростішу модель, якої достатньо для пояснення взаємозв'язку між змінними wage та age. Використаємо функцію `anova()`, яка виконує дисперсійний аналіз (ANOVA, за допомогою F-тесту) з метою перевірки нульової гіпотези про те, що моделі M1 достатньо для пояснення даних проти альтернативної гіпотези про необхідність застосування більш складної моделі M2. Щоб використовувати функцію `anova()`, M1 і M2 повинні бути вкладеними моделями: предиктори в M1 повинні бути підмножиною предикторів в M2. Тому ми оцінюємо п'ять різних моделей і послідовно порівнюємо простішу модель з більш складною.

```
> fit.1=lm(wage~age ,data=Wage)
> fit.2=lm(wage~poly(age ,2) ,data=Wage)
> fit.3=lm(wage~poly(age ,3) ,data=Wage)
> fit.4=lm(wage~poly(age ,4) ,data=Wage)
> fit.5=lm(wage~poly(age ,5) ,data=Wage)
> anova(fit.1,fit.2,fit.3,fit.4,fit.5)
Analysis of Variance Table

Model 1: wage ~ age
Model 2: wage ~ poly(age, 2)
Model 3: wage ~ poly(age, 3)
Model 4: wage ~ poly(age, 4)
Model 5: wage ~ poly(age, 5)

  Res.Df   RSS Df Sum of Sq    F Pr(>F)
1  2998 5022216
2  2997 4793430  1    228786 143.59 <2e-16 ***
3  2996 4777674  1      15756  9.89 0.0017 **
4  2995 4771604  1       6070  3.81 0.0510 .
5  2994 4770322  1       1283  0.80 0.3697
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

р-значення, що порівнює лінійну модель 1 з квадратною моделлю 2, дорівнює по суті нулю ( $<10^{-15}$ ), що вказує на те, що лінійної моделі недостатньо. Так само р-значення порівняння квадратної моделі 2 з кубічною моделлю 3 дуже низьке (0,0017), тому квадратичної моделі також недостатньо. р-значення порівняння кубічного і поліному 4-го степеня, дорівнює приблизно 5%, тоді як використання полінома 5-го степеня виглядає непотрібним оскільки відповідне р-значення дорівнює 0,37. Отже, або кубічний, або поліном 4-го степеня забезпечують розумну відповідність даним, а використання моделей нижчого чи вищого порядків не віправдане. У цьому випадку замість використання функції `anova()`

ми могли б отримати ці р-значення більш стисло, використовуючи той факт, що функція `poly()` створює ортогональні поліноми.

```
> coef(summary(fit.5))
    Estimate Std. Error t value Pr(>|t|)
(Intercept) 111.70     0.7288 153.2780 0.0000e+00
poly(age, 5)1 447.07    39.9161 11.2002 1.491e-28
poly(age, 5)2 -478.32    39.9161 -11.9830 2.368e-32
poly(age, 5)3 125.52    39.9161   3.1446 1.679e-03
poly(age, 5)4 -77.91    39.9161  -1.9519 5.105e-02
poly(age, 5)5 -35.81    39.9161  -0.8972 3.697e-01
```

Отримані р-значення однакові, а квадрат t-статистики дорівнює F-статистиці від функції `anova()`

```
> (-11.983)^2
[1] 143.6
```

Однак метод ANOVA працює незалежно від того, чи ми використовували ортогональні поліноми; це також працює, коли в моделі є й інші змінні. Наприклад, ми можемо використовувати `anova()` для порівняння цих трьох моделей:

```
> fit.1=lm(wage~education+age,data=Wage)
> fit.2=lm(wage~education+poly(age,2),data=Wage)
> fit.3=lm(wage~education+poly(age,3),data=Wage)
> anova(fit.1,fit.2,fit.3)
```

Як альтернативу використанню тестуванню гіпотез та ANOVA ми могли б вибрати поліноміальний степінь із використанням перехресної перевірки. Розглянемо далі проблему передбачення, чи заробляє людина більше ніж 250 000 доларів на рік. Ми продовжуємо, як і раніше, за винятком того, що спочатку ми створили відповідний вектор значень залежності змінної, а потім застосували функцію `glm()` з параметром `family = "binomial"` для того, щоб оцінити поліноміальну логістичну регресійну модель.

```
> fit=glm(I(wage>250)~poly(age,4),data=Wage,family=binomial)
```

Ми знову використовуємо `I()` для створення бінарної залежності змінної на льоту. Вираз `wage > 250` повертає логічне значення TRUE або FALSE, які `glm()` перетворує на двійкові, встановлюючи TRUE 1, а FALSE 0. Будуємо прогнози за допомогою функції `predict()`.

```
> preds=predict(fit,newdata=list(age=age.grid),se=T)
```

Побудова довірчих інтервалів трохи більше заплутана, ніж у випадку лінійної регресії. Тип передбачення за замовчуванням для моделі `glm()` є `type = "link"`, який ми тут використовуємо. Це означає, що ми отримуємо прогнози для `logit`: тобто ми оцінили модель вигляду

$$\log\left(\frac{P(Y=1|X)}{1-P(Y=1|X)}\right) = X\beta$$

і передбачення мають вигляд  $X\hat{\beta}$ . Стандартні відхилення також мають подібний вигляд. Для того, щоб побудувати інтервал довіри для  $P(Y=1|X)$ , ми використовуємо перетворення

$$P(Y=1|X) = \frac{\exp(X\beta)}{1+\exp(X\beta)}$$

```
> pfit=exp(preds$fit)/(1+exp(preds$fit))
> se.bands.logit = cbind(preds$fit+2*preds$se.fit, preds$fit-2*
  preds$se.fit)
> se.bands = exp(se.bands.logit)/(1+exp(se.bands.logit))
```

Зверніть увагу, що ми могли безпосередньо обчислити ймовірності, вибрали параметр `type = "response"` у функції `predict()`.

```
> preds=predict(fit,newdata=list(age=age.grid),type="response",
  se=T)
```

Однак відповідні інтервали довіри не були б правильними тому що ми в кінцевому результаті отримаємо від'ємні ймовірності! Побудуємо графік

```
> plot(age,I(wage>250),xlim=agelims,type="n",ylim=c(0,.2))
> points(jitter(age), I((wage>250)/5),cex=.5,pch="|",
  col="darkgrey")
> lines(age.grid,pfit,lwd=2, col="blue")
> matlines(age.grid,se.bands,lwd=1,col="blue",lty=3)
```

Ми намалювали вікові значення, що відповідають спостереженням із заробітною платою вищою ніж 250 як сірі позначки у верхній частині графіку, а ті, що відповідають заробітній платі нижчій за 250 - сірими позначками внизу графіка. Ми використали функцію `jitter()`, щоб змішати значення віку, щоб спостереження з однаковим віковим значенням не закривали одне одного. Для того, щоб пристосувати східчасту функцію використовуємо функцію `cut()`.

```

> table(cut(age, 4))
(17.9,33.5] (33.5,49] (49,64.5] (64.5,80.1]
    750       1399      779       72
> fit=lm(wage~cut(age, 4), data=Wage)
> coef(summary(fit))
              Estimate Std. Error t value Pr(>|t|)
(Intercept)     94.16     1.48   63.79 0.00e+00
cut(age, 4)(33.5,49]  24.05     1.83   13.15 1.98e-38
cut(age, 4)(49,64.5]  23.66     2.07   11.44 1.04e-29
cut(age, 4)(64.5,80.1]  7.64     4.99    1.53 1.26e-01

```

Тут `cut()` автоматично підібрав точки 33,5, 49 та 64,5 років. Ми також могли вказати наші власні граничні точки безпосередньо, використовуючи опцію `breaks`. Функція `cut()` повертає впорядковану категоріальну змінну. Потім функція `lm()` створює набір фіктивних змінних для використання в регресії. Категорія `age<33,5` залишається осторонь, тому коефіцієнт `beta0`, що становить 94,160 доларів США можна інтерпретувати як середню заробітну плату для осіб до 33,5 років віку, а інші коефіцієнти можна інтерпретувати як середні додаткові зарплати для тих, хто входить до інших вікових груп. Ми можемо будувати прогнози і графіки так само, як це було зроблено у випадку поліноміальної регресії.

## *Сплайни*

Для того, щоб застосувати регресійні сплайни, ми використовуємо бібліотеку `splines`. Регресійні сплайни можна пристосувати, побудувавши відповідну матрицю базисних функцій. Функція `bs()` генерує всю матрицю базисних функцій для сплайнів із заданим набором вузлів. За замовчуванням використовуються кубічні сплайни. Пристосуємо `wage` до `age` за допомогою регресійного сплайна.

```

> library(splines)
> fit=lm(wage~bs(age, knots=c(25,40,60)), data=Wage)
> pred=predict(fit, newdata=list(age=age.grid), se=T)
> plot(age, wage, col="gray")
> lines(age.grid, pred$fit, lwd=2)
> lines(age.grid, pred$fit+2*pred$se, lty="dashed")
> lines(age.grid, pred$fit-2*pred$se, lty="dashed")

```

Ми задали вузли у віці 25, 40 і 60 років, отже отримали сплайн із шістьма базовими функціями. Ми також можемо використовувати параметр df для використання сплайну із вузлами у рівномірно розташованих квантилях.

```
> dim(bs(age, knots=c(25,40,60)))
[1] 3000      6
> dim(bs(age, df=6))
[1] 3000      6
> attr(bs(age, df=6), "knots")
 25%   50%   75%
33.8  42.0  51.0
```

У цьому випадку вибрано вузли у віці 33,8, 42,0 та 51,0, що відповідають 25-му, 50-му та 75-му процентилям. Функція bs () також має аргумент degree, тому ми можемо використати сплайн будь-якого степеня, а не лише за замовчуванням 3-ого.

Для того, щоб замість цього використати природний сплайн, ми застосуємо функцію ns (). Використаємо природний сплайн з чотирма ступенями свободи.

```
> fit2=lm(wage~ns(age,df=4),data=Wage)
> pred2=predict(fit2,newdata=list(age=age.grid),se=T)
> lines(age.grid, pred2$fit,col="red",lwd=2)
```

Як і у випадку функції bs (), ми могли б замість цього вказати вузли безпосередньо за допомогою опції knots.

Для використання згладжувального сплайну, застосуємо функцію smooth.spline () .

```
> plot(age,wage,xlim=agelims,cex=.5,col="darkgrey")
> title("Smoothing Spline")
> fit=smooth.spline(age,wage,df=16)
> fit2=smooth.spline(age,wage,cv=TRUE)
> fit2$df
[1] 6.8
> lines(fit,col="red",lwd=2)

> lines(fit2,col="blue",lwd=2)
> legend("topright",legend=c("16 DF","6.8 DF"),
  col=c("red","blue"),lty=1,lwd=2,cex=.8)
```

Під час первого виклику smooth.spline () ми вказали df = 16. Потім функція визначає, яке значення  $\lambda$  веде до 16 ступенів свободи. В другому виклику smooth.spline (), ми вибираємо рівень гладкості шляхом перехресної перевірки; це призводить до значення  $\lambda$ , що дає 6,8 ступеня свободи.

Для використання локальної регресії, використовуємо функцію loess().

```
> plot(age, wage, xlim=agelims, cex=.5, col="darkgrey")
> title("Local Regression")
> fit=loess(wage~age, span=.2, data=Wage)
> fit2=loess(wage~age, span=.5, data=Wage)
> lines(age.grid, predict(fit, data.frame(age=age.grid)),
  col="red", lwd=2)
> lines(age.grid, predict(fit2, data.frame(age=age.grid)),
  col="blue", lwd=2)
> legend("topright", legend=c("Span=0.2", "Span=0.5"),
  col=c("red", "blue"), lty=1, lwd=2, cex=.8)
```

Тут ми виконали локальну лінійну регресію з використанням інтервалів 0,2 і 0,5, тобто кожен окіл складається з 20% або 50% спостережень. Чим більший інтервал, тим плавніше прилягання. Також можна використати бібліотеку locfit для використання моделей локальної регресії.

### Узагальнені адитивні моделі

Продемонструємо використання УАМ для прогнозування wage, використовуючи природні сплайні для функцій змінних year та age, та трактуючи education як якісну змінну. Оскільки ми отримуємо в результаті велику модель лінійної регресії з використанням відповідного набору базисних функцій, ми можемо оцінити її за допомогою функції lm().

```
> gam1=lm(wage~ns(year, 4)+ns(age, 5)+education, data=Wage)
```

Використаємо тепер згладжувальні сплайні, а не природні. Для того, щоб використовувати більш загальні типи УАМ, використовуючи згладжувальні сплайні або інші компоненти, які не можна виразити через базові функції та оцінити ці моделі за допомогою найменших квадратів, нам потрібно використати бібліотеку gam.

Для використання згладжувальних сплайнів застосуємо функцію s(), з бібліотеки gam. Вказуємо, що функція від year повинна мати 4 ступені свободи, а функція від age - 5 ступенів свободи. Оскільки education якісна змінна, ми залишаємо її такою, як є і вона перетворюється на чотири фіктивні змінні. Ми використовуємо функцію gam() щоб оцінити УАМ, використовуючи ці дані. Усі складові

оцінюються одночасно, беручи одна одну до уваги для пояснення залежності змінної.

```
> library(gam)
> gam.m3=gam(wage~s(year,4)+s(age,5)+education,data=Wage)
```

Для побудови графіка викликаємо функцію plot ():

```
> par(mfrow=c(1,3))
> plot(gam.m3, se=TRUE, col="blue")
```

Функція plot () розпізнає, що gam2 є об'єктом класу gam, і викликає відповідний метод plot.gam (). Незважаючи на те, що gam1 не є класу gam, а швидше класу lm, ми все-одно можемо використовувати plot.gam () .

```
> plot.gam(gam1, se=TRUE, col="red")
```

Тут ми змушені були використовувати plot.gam (), а не plot (). На цих графіках функція від year виглядає досить лінійною. Ми можемо виконати серію тестів ANOVA для того, щоб визначити, яка з цих трьох моделей є найкращою: УАМ, що виключає year (M1); УАМ, яка використовує лінійну функцію від year (M2); УАМ, що використовує сплайн-функцію від year (M3).

```
> gam.m1=gam(wage~s(age,5)+education,data=Wage)
> gam.m2=gam(wage~year+s(age,5)+education,data=Wage)
> anova(gam.m1,gam.m2,gam.m3,test="F")
Analysis of Deviance Table

Model 1: wage ~ s(age, 5) + education
Model 2: wage ~ year + s(age, 5) + education
Model 3: wage ~ s(year, 4) + s(age, 5) + education
Resid. Df Resid. Dev Df Deviance    F   Pr(>F)
1      2990     3711730
2      2989     3693841  1      17889  14.5  0.00014 ***
3      2986     3689770  3       4071   1.1  0.34857
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1
```

Ми виявили, що є вагомі докази того, що УАМ з лінійною функцією від year краща, ніж УАМ, яка взагалі не включає year (р-значення = 0,00014). Однак немає жодних доказів необхідності використання нелінійної функції від year (р-значення = 0,349). Отже, M2 є кращою моделлю.

Функція summary () наводить підсумки оцінки УАМ.

```

> summary(gam.m3)

Call: gam(formula = wage ~ s(year, 4) + s(age, 5) + education,
  data = Wage)
Deviance Residuals:
    Min      1Q  Median      3Q     Max 
-119.43 -19.70 -3.33  14.17 213.48 

(Dispersion Parameter for gaussian family taken to be 1236)

Null Deviance: 5222086 on 2999 degrees of freedom
Residual Deviance: 3689770 on 2986 degrees of freedom
AIC: 29888

Number of Local Scoring Iterations: 2

DF for Terms and F-values for Nonparametric Effects

          Df Npar Df Npar F Pr(F)
(Intercept) 1
s(year, 4)   1      3     1.1   0.35
s(age, 5)    1      4    32.4 <2e-16 ***
education    4
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ',' 1

```

р-значення для year та age відповідають нульовим гіпотезам про лінійний зв'язок проти альтернативних - нелінійний зв'язок. Високе р-значення для year підтверджує наш висновок з тесту ANOVA, що лінійна функція є коректним вибором для цієї змінної. Однак є дуже чіткі докази, що для age потрібна нелінійна функція. Ми можемо будувати прогнози на основі gam об'єктів, як і з об'єктів lm, за допомогою методу predict () для класу gam. Тут ми робимо прогнози для навчального набору.

```
> preds=predict(gam.m2,newdata=Wage)
```

Ми також можемо використовувати локальну регресію як будівельні блоки в УАМ, використовуючи функцію lo () .

```

> gam.lo=gam(wage~s(year,df=4)+lo(age,span=0.7)+education,
  data=Wage)
> plot.gam(gam.lo, se=TRUE, col="green")

```

Ми використали локальну регресію для age, з інтервалом 0,7. Ми також можемо використовувати функцію lo () для створення змінних взаємодій перед викликом функції gam (). Наприклад,

```
> gam.lo.i=gam(wage~lo(year,age,span=0.5)+education,  
+ data=Wage)
```

відповідає моделі, в якій перший член описує взаємодію між year та age, пристосованій на основі локальної регресійної поверхні. Ми можемо побудувати графік отриманої двовимірної поверхні, якщо спочатку встановити пакет akima.

```
> library(akima)  
> plot(gam.lo.i)
```

Для того, щоб використати логістичну регресію УАМ, ми знову використовуємо функцію I () при побудові двійкової залежності змінної і встановлюємо family=binomial.

```
> gam.lr=gam(I(wage>250)~year+s(age,df=5)+education,  
+ family=binomial,data=Wage)  
> par(mfrow=c(1,3))  
> plot(gam.lr,se=T,col="green")
```

Неважко помітити, що в категорії <HS немає людей з високим рівнем доходу:

```
> table(education,I(wage>250))
```

| education          | FALSE | TRUE |
|--------------------|-------|------|
| 1. < HS Grad       | 268   | 0    |
| 2. HS Grad         | 966   | 5    |
| 3. Some College    | 643   | 7    |
| 4. College Grad    | 663   | 22   |
| 5. Advanced Degree | 381   | 45   |

Таким чином, ми підходимо до логістичної регресії УАМ, використовуючи всі категорії, крім цієї. Це забезпечує кращі результати.