Odhad funkcie dvoch premenných

Tomáš Bacigál

2023-08-25

Table of contents

Zadanie

Na základe pozorovaní nájsť matematický predpis aproximujúci vzťah parametrov (okna): vysvetľujúcich veličín VDF (x), ρ (y) a odozvy D/ADF (f)

Vstupné údaje

```
library(ggplot2)
dat <- read.table("data.txt", header = TRUE, dec = ",") |>
  tidyr::pivot_longer(-1, names_to = "y", values_to = "f") |>
  dplyr::mutate(y = as.numeric(substring(y, 2))) |>
  dplyr::rename(x = VDF) |>
  dplyr::arrange(y, x)
```

Súbor údajov rozdelený do troch tabuliek podľa hodnôt y:

```
split(dat, f = dat$y) |>
  purrr::walk(function(x) print(knitr::kable(x)))
```

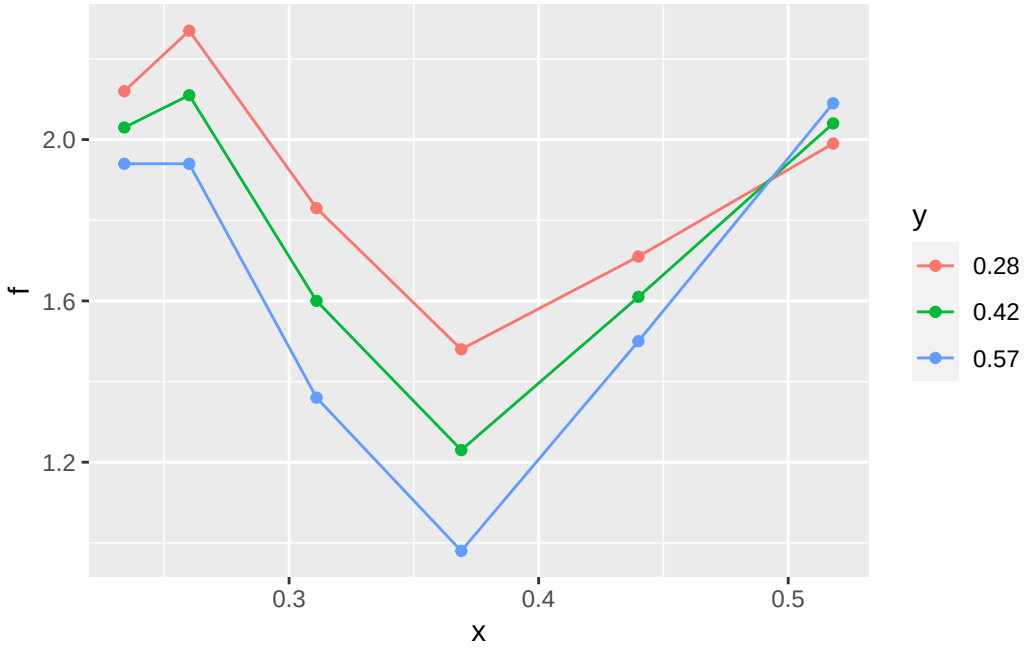
x	y	f
0.234	0.28	2.12
0.260	0.28	2.27
0.311	0.28	1.83
0.369	0.28	1.48
0.440	0.28	1.71
0.518	0.28	1.99

x	y	f
0.234	0.42	2.03
0.260	0.42	2.11
0.311	0.42	1.60
0.369	0.42	1.23
0.440	0.42	1.61
0.518	0.42	2.04

x	y	f
0.234	0.57	1.94
0.260	0.57	1.94
0.311	0.57	1.36
0.369	0.57	0.98
0.440	0.57	1.50
0.518	0.57	2.09

```
dat_p <- dat |> dplyr::mutate(y = factor(y))

dat_p |>
  ggplot() + aes(x = x, y = f, color = y, group = y) +
  geom_line() + geom_point()
```



Identifikácia tvaru funkcie a odhad parametrov

Hľadaná funkcia môže mať v rezoch zodpovedajúcich úrovniam premennej y približne parabolický tvar, pričom tieto rezy nie sú pozdĺž osi x v konštantnej vzdialenosti (vertikálne). Preto pátranie po ideálnej regresnej funkcii začneme parametrickou triedou

$$F_1(x) = a_{11}x^2 + a_{12}xy + a_{22}y^2 + a_1x + a_2y + a_0$$

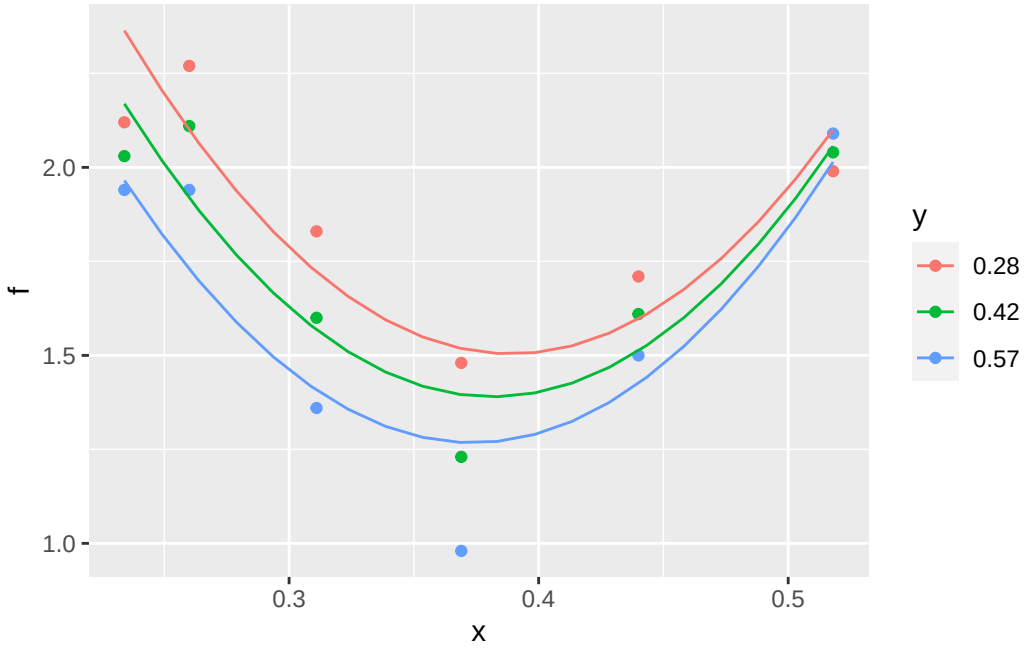
kde a_{ij} sú parametre. Keďže ide o funkciu lineárnu v parametroch, ich odhad sa vykoná bežnou metódou najmenších štvorcov (OLS).

```
fit1 <- lm(f ~ I(x^2) + I(y^2) + x * y, dat)
fit1$coefficients |> round(3)
```

```
(Intercept)      I(x^2)      I(y^2)           x           y         x:y
      7.570      35.795       0.099     -28.914      -2.350       3.806
```

```
grid <- expand.grid(x = modelr::seq_range(dat$x, n = 20),
                    y = unique(dat$y), KEEP.OUT.ATTRS = FALSE)
grid1 <- grid |>
  dplyr::mutate(f = predict(fit1, newdata = dplyr::pick(everything())),
                y = factor(y))

ggplot() + aes(x = x, y = f, color = y, group = y) +
  geom_point(data = dat_p) +
  geom_line(data = grid1)
```



Tieto rezy by sa podľa pozorovaných hodnôt mali niekde v hornom spektre premennej x pretínať, preto interakciu medzi prediktormi posilníme pridaním ďalšieho interakčného členu, yx^2,

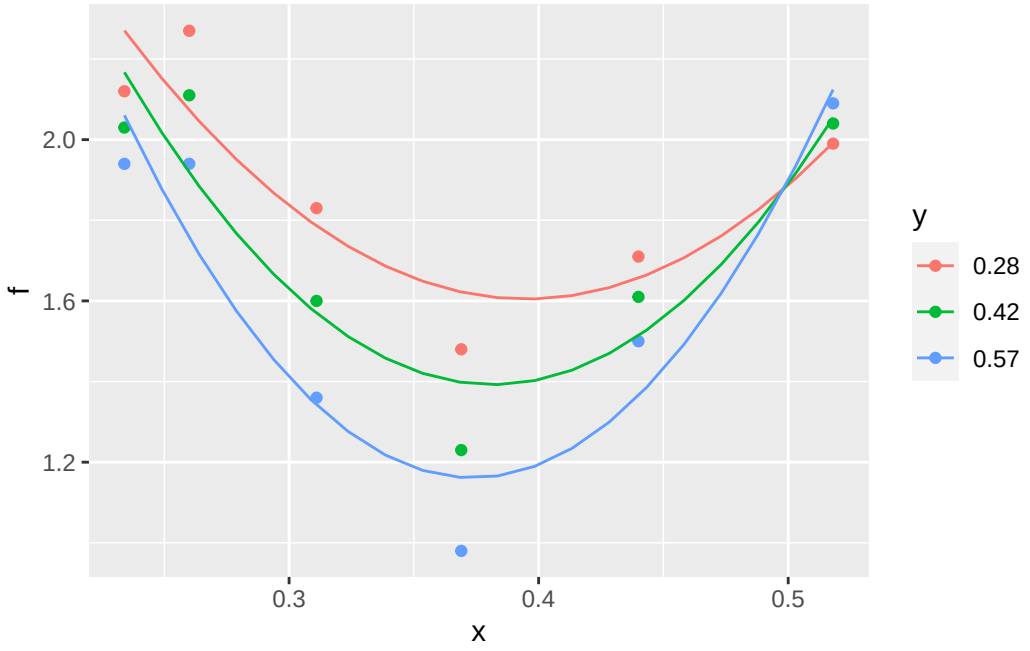
$$F_2(x, y) = a_{11,2}x^2y + a_{11}x^2 + a_{12}xy + a_{22}y^2 + a_1x + a_2y + a_0$$

```
fit2 <- update(fit1, "~ . + I(y*x^2)")
fit2$coefficients |> round(3)
```

```
(Intercept)      I(x^2)      I(y^2)           x           y  I(y * x^2)
      3.701       5.870       0.099      -6.555       6.790      70.690
        x:y
    -49.010
```

```
grid2 <- grid |>
  dplyr::mutate(f = predict(fit2, newdata = dplyr::pick(everything())),
                y = factor(y))

ggplot() + aes(x = x, y = f, color = y, group = y) +
  geom_point(data = dat_p) +
  geom_line(data = grid2)
```



Rezy skutočnej regresnej funkcie však vôbec nemusia mať tvar paraboly. Skúsime preto aj po častiach lineárnu parametrickú triedu funkcií (lineárne regresné splajny). Využijeme pritom pozorovanie, že závislosť medzi x a f sa významne mení v blízkom okolí bodu $x = k_1 = 0.369$. Funkcia bude mať tvar

$$\begin{aligned} F_3(x,y) &= b_1 + b_2 x + b_3 \max(0, x - k_1) + b_4 xy + b_5 \max(0, x - k_1) y \\ &= \begin{cases} b_1 + b_2 x + b_4 xy & \text{ak } x \le k_1 \\ (b_1 - b_3 k_1) + (b_2 + b_3) x - b_5 k_1 y + (b_4 + b_5) xy & \text{inak.} \end{cases} \end{aligned}$$

kde k_1 sa nazýva uzol (knot) a určuje sa ešte pred odhadom parametrov b_i.

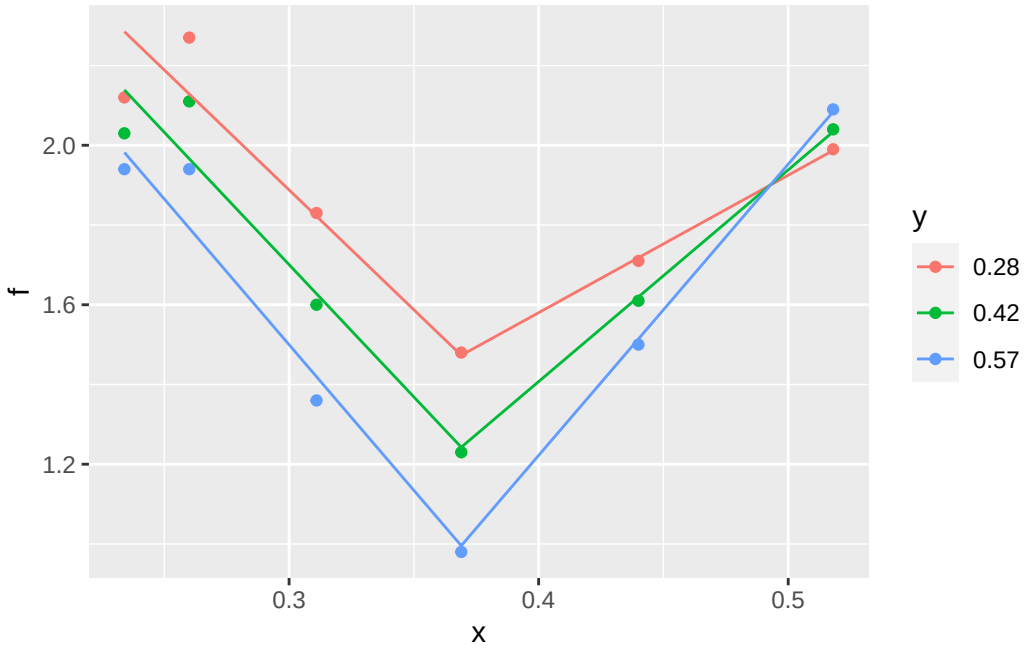
```
knot <- 0.369
fit3 <- dat |>
  dplyr::mutate(max1 = pmax(0, x-knot)) |>
  lm(f ~ (x + max1) * y - y, data = _)
fit3$coefficients |> round(3)
```

```
(Intercept)           x        max1         x:y      max1:y
      3.692      -4.763       4.489      -4.462      17.757
```

```
grid <- expand.grid(x = c(min(dat$x), knot, max(dat$x)),
                    y = unique(dat$y), KEEP.OUT.ATTRS = FALSE) |>
  dplyr::mutate(max1 = pmax(0, x-knot))

grid3 <- grid |>
  dplyr::mutate(f = predict(fit3, newdata = dplyr::pick(everything())),
                y = factor(y))

ggplot() + aes(x = x, y = f, color = y, group = y) +
  geom_point(data = dat_p) +
  geom_line(data = grid3)
```



Vidno, že model k dátam prilieha tesnejšie než v predošlých prípadoch a pritom je úspornejší (má menej parametrov a teda i stupňov voľnosti). Pridanie ďalšieho uzla je možné, no zvýši komplexnosť modelu o 2 parametre, preto je vždy vhodné racionálne posúdiť praktickú využiteľnosť takého kroku (zvoliť kompromis medzi adaptabilitou a interpretovateľnosťou modelu).

$$F_4(x,y) = b_1 + b_2 x + b_3 \max(0, x-k_1) + b_4 \max(0, x-k_2) + b_5 xy + b_6 \max(0, x-k_1)y + b_7 \max(0, x-k_2)y$$

Uvažujme uzol $k_2 = 0.26$.

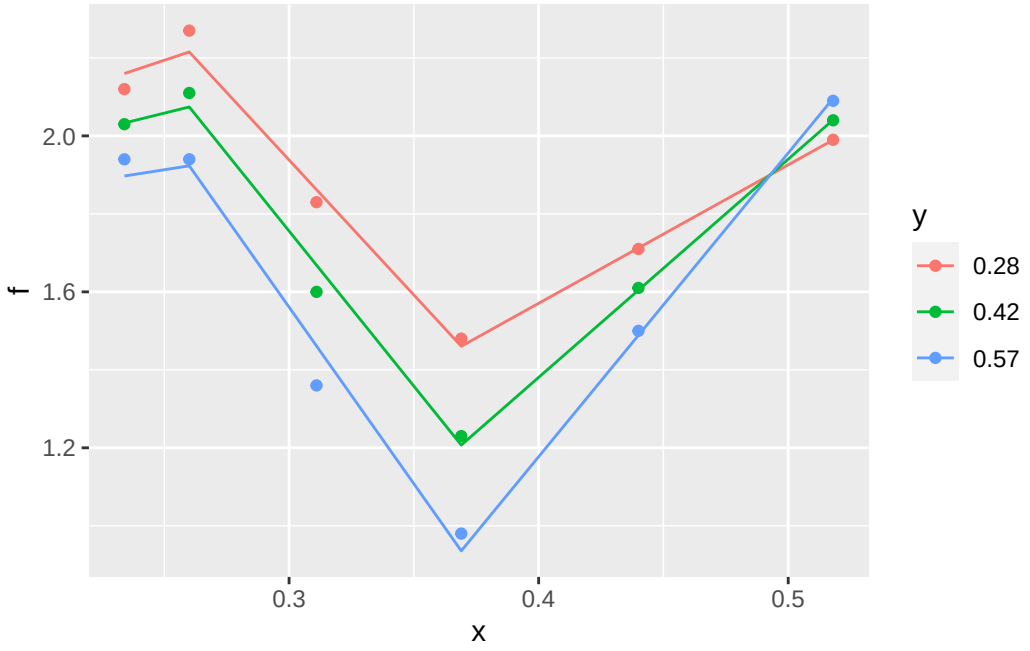
```r
knots <- c(0.369, 0.26)
fit4 <- dat |>
  dplyr::mutate(max1 = pmax(0, x-knots[1]), max2 = pmax(0, x-knots[2])) |>
  lm(f ~ (x + max1 + max2 ) * y - y, data = _)
fit4$coefficients |> round(3)
```

```
(Intercept)           x        max1        max2         x:y      max1:y
      1.662       3.213       4.323      -8.076      -3.873      21.975
     max2:y
     -3.489
```

```r
grid <- expand.grid(x = c(min(dat$x), knots, max(dat$x)),
                    y = unique(dat$y), KEEP.OUT.ATTRS = FALSE) |>
  dplyr::mutate(max1 = pmax(0, x-knots[1]), max2 = pmax(0, x-knots[2]))

grid4 <- grid |>
  dplyr::mutate(f = predict(fit4, newdata = dplyr::pick(everything())),
                y = factor(y))

ggplot() + aes(x = x, y = f, color = y, group = y) +
  geom_point(data = dat_p) +
  geom_line(data = grid4)
```



Porovnanie a záver

Modely sa líšia adaptabilitou (vyjadrenou koeficientom determinácie R^2 korigovaným o počet parametrov) aj komplexnosťou (reprezentovanou stupňami voľnosti DF).

```
list(fit1, fit2, fit3, fit4) |> setNames(1:4) |>
  lapply(function(x) {
    x <- summary(x)
    cbind(R2adj = round(x$adj.r.squared,2), DF = x$df[1]) |> as.data.frame()
  }) |>
  dplyr::bind_rows() |>
  dplyr::mutate(model = 1:4, .before = 1) |>
  dplyr::mutate(type = rep(c("polynomial", "piecewise linear"), each=2), .after = -1) |>
  knitr::kable()
```

model	R2adj	DF	type
1	0.76	6	polynomial
2	0.80	7	polynomial
3	0.93	5	piecewise linear
4	0.98	7	piecewise linear

Je zjavná prevaha modelov druhého typu. Pomocou ANOVA je možné testovať štatistický rozdiel v popisnej schopnosti modelov rovnakého typu. Nulová hypotéza tvrdí, že jednoduchší model je postačujúci.

```
anova(fit1, fit2)
```

```
Analysis of Variance Table

Model 1: f ~ I(x^2) + I(y^2) + x * y
Model 2: f ~ I(x^2) + I(y^2) + x + y + I(y * x^2) + x:y
  Res.Df     RSS Df Sum of Sq     F Pr(>F)
1     12 0.36241
2     11 0.28177  1  0.080639 3.148 0.1037
```

```
anova(fit3, fit4)
```

```
Analysis of Variance Table

Model 1: f ~ (x + max1) * y - y
Model 2: f ~ (x + max1 + max2) * y - y
  Res.Df      RSS Df Sum of Sq      F    Pr(>F)
1     13 0.108613
2     11 0.027035  2  0.081578 16.596 0.0004767 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

V prípade polynomických modelov nemožno na hladine významnosti 5% nulovú hypotézu zamietnuť. Naopak, v prípade po častiach lineárnych modelov sa z popisného hladiska ako vhodnejší ukazuje ten zložitejší model (F_4).

Interpretačne je na tom najlepšie model F_3,

$$\begin{aligned} F_3(x,y) &= 3.7 + (-4.8)x + 4.5\max(0, x-0.37) + (-4.5)xy + 18\max(0, x-0.37) \\ &= \begin{cases} 3.7 + (-4.8)x + (-4.5)xy & \text{ak } x \leq 0.37 \\ 2.04 + (-0.3)x - 6.64y + 13.5xy & \text{inak.} \end{cases} \end{aligned}$$

z ktorého vyplýva, že so zvýšením x o Δx sa hodnota funkcie

- pre $x < 0.37$ zmení o $(b_2 + b_4 y)\Delta x$
  (teda napr. ak je $y = 0.3$, a x narastie o 0.1, potom funkčná hodnota klesne o 0.62),
- inak sa zmení o $(b_2 + b_3 + [b_4 + b_5]y)\Delta x$
  (teda opäť, ak je $y = 0.3$ a $\Delta x = 0.1$, potom hodnota F_3 narastie o 0.38).