Četrta domača naloga

Odkrivanje enačb

Napredno strojno učenje 2023

Dejan Perić

Vpisna številka: 27222015

14. junij 2023

1 Odkrivanje enačb in uporaba predznanja

1.1 Priprava podatkov in prva uporaba predznanja

Uvozimo `.csv` datoteko s podatki v `jupyter notebook` kot `pandas` tabelo. Uporabimo predznanje iz druge in tretje točke ter ustvarimo nov stolpec razlik temperatur Δ_T in zatem odstranimo stolpca T_w in T_a. Stolpca odstranimo zaradi predznanja, ker sklepamo, da absolutni vrednosti temperatur verjetno nista v enačbi in bosta verjetno zgolj motila proces učenja. Ker vidimo, da je temperatura v hladilnem stolpu zmeraj večja od temperature zunaj, določimo $\Delta_T = T_w - T_a$. To tabelo shranimo kot `podatki_delta`. Druga točka domenskega predznanja si dalje interpretiramo; verjetno bo razlika temperatur zapakirana v neko pozitivno naraščujočo funkcijo na intervalu $[0, \infty)$, ki najverjetneje ne bo linearna. Take funkcije bi lahko bili razni polinomi, eksponentna funkcija, logaritemska funcija, korenska funkcija in podobno.
Upoštevajoč prvo točko domenskega predznanja ustvarimo dodatno tabelo `podatki_sintheta`, kjer predpostavimo, da je sinusna funkcija neka naravna funkcija za uporabo v tem primeru, kjer ima toplotni tok najvišjo vrednost ko je $\theta = \frac{\pi}{2}$. Tu seveda predpostavimo da veter enako vpliva če piha navzgor ali pa navzdol glede na hladilni stolp. V tej tabeli dodamo nov stolpec $\sin(\theta)$. Seveda pa bomo uporabili tudi tabelo brez tega stolpca pri metodah, pri katerih lahko podamo razne funkcije (npr. pri metodi z gramtiko).
V podatkih opazimo, da je $\eta \in [0, 1]$. Domensko predznanje glede te spremenljivke nam pove, da bo η verjetno nastopal v neki monotoni padajoči

funkciji, kot na primer $1 - x$, x^{-1}, e^{-x} in podobno.
Kot funkcijo napake bomo v vseh primerih uporabili $err(f)$, ki se glasi

$$err(f) = \frac{1}{500} \sum_{k=1}^{500} (Q_k - f(\Delta_{T_k}, \theta_k, \eta_k))^2 .$$

1.2 Linearna regresija

Glede na to, da so meritve fizikalne narave, lahko predpostavljamo, da je v podatkih napaka oz nek šum. Zato verjetno metode linearne regresije ne bodo uspešne, tudi zaradi pomankanja dodatnih funkcij in operacij poleg polinomskih členov. Vseeno poskusimo. Kot na vajah tabeli `podatki_sintheta` s `PolynomialFeatures` iz knjižnice `sklearn.preprocessing` dodamo polinomske člene do tretje stopnje. Še najboljši rezultat dobimo z lasso regresijo, kjer nastavimo parametre $\lambda = 3$, meja $= 0.01$. Dobimo enačbo

$$\begin{aligned} Q &= 0.032 \cdot \eta \cdot \Delta_T + 0.033 \cdot \Delta_T \cdot \sin(\theta) + 0.021 \cdot \eta^2 \cdot \Delta_T \\ &= \Delta_T(0.032 \cdot \eta + 0.033 \cdot \sin(\theta) + 0.021 \cdot \eta^2), \end{aligned}$$

napaka te enačbe pa je $err = 0.71322$. Če z upoštevanjem predznanja zamenjamo stolpec η z $1 - \eta$, dobimo pri lasso regresiji z istimi parametri enačbo z manj napake ($err = 0.29765$). Enačba izgleda takole:

$$\begin{aligned} Q &= 0.033 \cdot \Delta_T + 0.085 \cdot \Delta_T \cdot \sin(\theta) \cdot (1 - \eta) + 0.016 \cdot \Delta_T \cdot (1 - \eta)^2 \\ &= \Delta_T(0.033 + 0.085 \cdot \sin(\theta) \cdot (1 - \eta) + 0.016 \cdot (1 - \eta)^2). \end{aligned}$$

Z dobljenimi enačbami ne moremo biti preveč zadovoljni, saj vidimo, da je toplotni tok linearno odvisen z razlikami temperatur, kar pa je nasprotno od domenskega predznanja, poleg tega pa tudi napaka ni tako majhna, da bi lahko bili zadovoljni.

1.3 Knjižnica `ProGED`

Poskusimo najti pravo enačbo z uporabo verjetnostnih gramatik. Upoštevajoč podatke v tabeli `podatki_delta` ustvarimo sledečo gramatiko.

```
E -> E '*' F [0.6] | F [0.4]
F -> F '+' K [0.1] | F '-' K [0.1] | K [0.8]
K -> T '(' M ')' [0.3] | 'C' [0.2] | V [0.5]
T -> 'sin' [1]
M -> 'theta' [1]
V -> 'deltaT' [0.6] | 'eta' [0.4]"
```

Slika 1: Verjetnostna gramatika za naš primer

To gramatiko uporabimo za generiranje modelov z metodo `EqDisco` iz knjižnice `ProGED`. Najprej nastavimo število primerov (`sample_size`) na 100. Nato z metodo `fit_models` pogledamo, kateri modeli imajo najmanjšo napako. Rezultate ipišemo z metodo `get_results`. Tu dobimo enačbo

$$Q = 0.0032669 \cdot \Delta_T^2 \cdot \sin(\theta)$$

z napako $err = 0.12511$. Napaka je manjša kot pri rezultatu, dobljenem z lasso regresijo. Vidimo, da Q ni linearno odvisen od Δ_T, vendar nas malo moti, da v enačbi ne nastopi izolacijski indeks η. Metoda je tu potrebovala približno 5 sekund za učenje. Pri povečanju primerov na 1000 primerov dobimo enačbo

$$\begin{aligned} Q &= -0.0030807 \cdot \Delta_T^2 \cdot \eta^2 \cdot \sin(\theta) + 0.0042041 \cdot \Delta_T^2 \cdot \sin(\theta) \\ &= 0.0030807 \cdot \Delta_T^2 \cdot \sin(\theta) \cdot (1.3647 - \eta^2). \end{aligned}$$

Tu je napaka enaka $err = 0.023551$, kar je že dosti boljše od prejšnje napake, poleg tega pa je tudi η vključen v enačbo. Metoda je tu potrebovala približno 40 sekund. Kot poizkus sem pognal metodo z milijon primeri. Tu je metoda potrebovala približno 7 ur, napaka pa je bila $err = 0.013306$, ne spet tako boljša od prejšnje gledano na povečano število primerov in čas delovanja. Enačba se tu glasi

$$Q = 0.0031737 \cdot \Delta_T^2 \cdot (1.5118 \cdot \sin(\theta) - \eta \cdot \sin(\theta) - 0.0065611).$$

1.4 Knjižnica `pySR`

Poskusimo najti enačbo še z metodo `PySRRegressor` iz knjižnice `pySR`. Uporabili bomo podatke iz tabele `podatki_delta`.

```
model = PySRRegressor(
    niterations=40,
    binary_operators=["+", "*", "-", "/"],
    unary_operators=["exp", "sin", "cos", "sinh", "log", "sqrt", "cosh", "tan"],
    extra_sympy_mappings={"inv": lambda x: 1 / x},
    loss="loss(prediction, target) = (prediction - target)^2"
)
```

Slika 2: Generiranja modela z metodo `PySRRegressor`

Pri definiranju modela je treba podati, katere binarne operacije in funkcije hočemo, da model uporabi in kako naj bo definirana funkcija napake. Ko naše podatke naučimo na tem modelu, nam model vrne več možnih funkcij naraščajočih po kompleksnosti in nam potem glede na napako in kompleksnost enačbe vrne najbolj ustrezno enačbo. V tem primeru smo dobili enačbo

$$0.0052073 \cdot \Delta_T^2 \cdot e^{-\eta} \sin(\theta).$$

Napaka tu znaša $err = 0.015174$, kar je podobna napaka kot pri zadnjih primerih z gramatiko. Podatki so se na modelu učili približno 20 sekund. V prejšnjih poglavjih smo z upoštevanjem predznanja privzeli, da se θ preslika s sinusno funkcijo. Tu tega nismo privzeli, vendar vidimo, da se tu model izmed več možnih podanih funkcij odloči prav tako za sinusno funkcijo. Struktura enačbe se v primerjavi z enačbami dobljene z verjetnostnimi gramatikami razlikuje le v členu η. Tu se ta člen preslika z eksponentno funkcijo. Poskusimo še enkrat naučiti naše podatke na modele, le da tu za možne funkcije podamo le sinusno funkcijo.

$$Q = 0.0031755 \cdot \Delta_T^2 \cdot (1.5030 - \eta) \cdot \sin(\theta)$$

Dobljena enačba ima napako $err = 0.013328$, kar je manjša napako kot napaka prejšnje enačbe in ravno malo večja kot zadnja enačba dobljena s knjižnico `ProGED` (ki pa je trajala 7 ur). Podatki so se tu na modelu učili 15 sekund.

1.5 Zaključek

Ugotovili smo, da linearna regresija pri iskanju naše enačbe ni bila tako dobra metoda kot ostali dve. Metodi `ProGED` in `pySR` nam podata nekaj par potencialnih končnih enačb z napako stopnje 10^{-2}. Napake manjšega reda nismo dobili pri nobeni metodi, tako da se lahko vprašamo, ali je bilo sploh možno dobiti napako manjšega reda. Možno je, da je ta napaka posledica šuma pri meritvah podatkov. Na koncu se odločimo za drugo enačbo dobljeno

z metodo `pySR` ($Q = 0.0031755 \cdot \Delta_T^2 \cdot (1.5030 - \eta) \cdot \sin(\theta)$), ki ji glede na napake istega reda preostalih dobljenih enačb ne smemo preveč zaupati. Smo pa lahko dokaj prepričani, da v enačbi spremenljivki Δ_t in θ nastopata ena v kvadratni druga pa v sinusni funkciji.