Uvod v odkrivanje enačb in simbolno regresijo

Ljupčo Todorovski

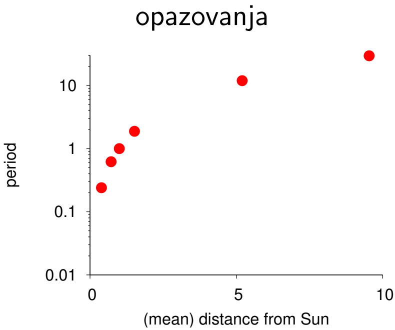
Univerza v Ljubljani, Fakulteta za matematiko in fiziko
Institut Jožef Stefan, Odsek za tehnologije znanja (E8)

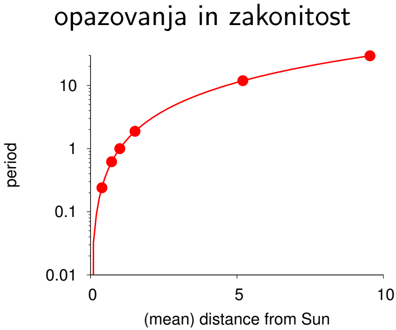
April 2023

Odkrivanje enačb: tretji Keplerjev zakon

Rekonstrukcija Keplerjevega tretjega zakona iz podatkov

$$d^3/p^2 = const$$





Pregled predavanja

1. Motivacija
   - Uvod v odkrivanje enačb in simbolno regresijo
   - Razmerje do drugih nalog
   - Dekompozicija naloge in osnovni pristopi
2. Evolucijski pristop
   - Kromosomi za odkrivanje enačb
   - Evolucijski operatorji nad izrazi
   - Uspešnost kromosoma pri odkrivanju enačb
   - Evolucijska optimizacija
   - Težave s preprileganjem
3. Odkrivanje enačb s propozicionalizacijo

Definicija naloge

Za podani par

- Podatkovne množice S: $X_i : D_{X_i} = \mathbb{R}, i = 1, 2, \ldots, p$; $Y : D_Y = \mathbb{R}$
- Prostora $\mathcal{E}$ matematičnih izrazov $E(X)$ iz spremenljivk X

Najdi enačbo oblike $Y = E(X)$ za katero

$$\min_{E(X) \in \mathcal{E}} \sum_{(\boldsymbol{x}, y) \in S} (y - E(\boldsymbol{x}))^2$$

Odkrivanje tretjega Keplerjevega zakona

Podana podatkovna množica S

planet ID	$X_1 = p$	$Y = d$
Merkur	0.389	87.77
Venera	0.724	224.7
Zemlja	1	365.25
Mars	1.524	686.95
Jupiter	5.2	4332.62
Saturn	9.51	10759.2

Najdi enačbo

$$d = \sqrt[3]{7.496 \cdot 10^{-6} \cdot p^2}; \qquad d^3/p^2 = 7.496 \cdot 10^{-6}$$

Razmerje z linearno regresijo

Simbolna regresija je nelinearna regresija

- Pri linearni regresiji je ciljna enačba oblike $Y = c_0 + \sum_{i=1}^{p} c_i \cdot X_i$
- Pri simbolni regresiji pa $Y = E(X)$, $E(X)$ je poljubne oblike

Včasih možna ročna transformacija

- Vpeljemo nove spremenljivke $\log p$ in $\log d$
- Ciljna enačba med novimi spremenljivkami linearne oblike
- $\log d = \frac{2}{3} \log p + \frac{1}{3} \log 7.496 \cdot 10^{-6}$

Razmerje s strojnim učenjem

Odkrivanje enačb je posebna vrsta strojnega učenja

- Prostor možnih modelov so enačbe (za razliko od dreves, najbližjih sosedov ali nevronskih mrež)
- Rezultat simbolne regresije naj bi bil bolj razumljiv
- Enačbe so standarden in dobro uveljavljen formalizem v znanosti

Tudi tukaj možna pretvorba

- Posebne vrste nevronov v umetnih nevronskih mrežah
- Opravljajo aritmetične operacije namesto običajne obtežene vsote
- Glej npr. model nevronske mreže EQL

Hevristični pristop Bacon

Tri preproste hevristike

1. Uvajanje nove spremenljivke
   Če spremenljivka U narašča kadarkoli V pada, uvedi $U \cdot V$
2. Uvajanje nove spremenljivke
   Če spremenljivka U narašča kadarkoli V narašča, uvedi U/V
3. Ugotavljanje invariante
   Če ima spremenljivka U nizko varianco, zastavi enačbo $U = c$

Bacon: Odkrivanje tretjega Keplerjevega zakona (1)

planet ID	$X_1 = p$	$Y = d$
Merkur	0.389	87.77
Venera	0.724	224.7
Zemlja	1	365.25
Mars	1.524	686.95
Jupiter	5.2	4332.62
Saturn	9.51	10759.2

Kadarkoli narašča p narašča tudi d: uvedi d/p.

Bacon: Odkrivanje tretjega Keplerjevega zakona (2)

planet ID	d	p	d/p
Merkur	0.389	87.77	4.43E-03
Venera	0.724	224.7	3.22E-03
Zemlja	1	365.25	2.74E-03
Mars	1.524	686.95	2.22E-03
Jupiter	5.2	4332.62	1.20E-03
Saturn	9.51	10759.2	8.84E-04

Kadarkoli narašča d, d/p pada: uvedi $d \cdot d/p = d^2/p$.

Bacon: Odkrivanje tretjega Keplerjevega zakona (3)

planet ID	d	p	d/p	d^2/p
Merkur	0.389	87.77	4.43E-03	1.72E-03
Venera	0.724	224.7	3.22E-03	2.33E-03
Zemlja	1	365.25	2.74E-03	2.74E-03
Mars	1.524	686.95	2.22E-03	3.38E-03
Jupiter	5.2	4332.62	1.20E-03	6.24E-03
Saturn	9.51	10759.2	8.84E-04	8.41E-03

Kadarkoli narašča d/p, d^2/p pada: uvedi $d/p \cdot d^2/p = d^3/p^2$.

Bacon: Odkrivanje tretjega Keplerjevega zakona (4)

planet ID	d	p	d/p	d^2/p	d^3/p^2
Merkur	0.389	87.77	4.43E-03	1.72E-03	7.64E-06
Venera	0.724	224.7	3.22E-03	2.33E-03	7.52E-06
Zemlja	1	365.25	2.74E-03	2.74E-03	7.50E-06
Mars	1.524	686.95	2.22E-03	3.38E-03	7.50E-06
Jupiter	5.2	4332.62	1.20E-03	6.24E-03	7.49E-06
Saturn	9.51	10759.2	8.84E-04	8.41E-03	7.43E-06

Varianca d^3/p^2 je nizka ($< 10^{-14}$), vzpostavi enačbo $d^3/p^2 = 7.51 \cdot 10^{-6}$.

Običajna dekompozicija naloge

Dva (prepletena) koraka odkrivanja enačb

Iskanje ustrezne strukture enačbe

- $d^3/p^2 = c$ ali $F = m \cdot g$
- To je problem kombinatorične optimizacije

Ocenjevanje vrednosti parametrov

- $c = 7.496 \cdot 10^{-6}$ ali $g = 9.81$
- To je problem numerične optimizacije

Splošni pristop ustvari-in-preizkusi, *generate-and-test*

Require: S je učna podatkovna množica, spremenljivke $Y, X_i, i = 1 \dots p$

Ensure: e je enačba oblike $Y = E(X)$

function GenerateAndTest(S)
  $CurrentError = \infty$
  while ($E(X)$ = Generate(X)) != NULL **do**
    $Error$ = EstimateParameters($E(X), S$)
    if $Error < CurrentError$ **then**
      $e = (Y = E(X))$
    end if
  end while
  return e
end function

Preizkus strukture $E(X)$: EstimateParameters

Za podano enačbo $Y = E(X)$ in množico S

- Z neznanimi vrednostmi m-tih parametrov $\boldsymbol{c} = (c_1, c_2, \ldots c_m) \in \mathbb{R}^m$
- Najdi optimalne vrednosti $\boldsymbol{c}^*$, tako da velja

$$\boldsymbol{c}^* = \arg\min_{c \in \mathbb{R}^m} \|Y - E(X, \boldsymbol{c})\|$$

- $\|Y - E(X, \boldsymbol{c})\|$ izračunamo na primerih iz S

Običajni problem numerične optimizacije: velika izbira algoritmov. V okviru tega predmeta (pri meta učenju), smo obravnavali Bayesovo optimizacijo.

Ustvarjanje strukture $E(X)$: Generate

Več možnosti

1. Stohastični evolucijski pristop (običajna, široko uporabljena možnost)
2. Sistematični pristop: naštevanje vseh možnosti
3. Sistematični pristop: operator izostritve

V nadaljevanju predavanj pregled teh pristopov.

Osnovna ideja

Kromosomi za predstavitev objektov

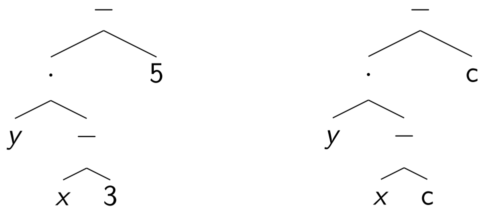
- Osnovni kromosomi: Boolovi vektorji
- Za enačbe: drevesna predstavitev matematičnih izrazov

Nastanek novih kromosomov

- Evolucijski operatorji križanja in mutacije
- Izbira kromosomov na osnovi funkcije uspeha, *fitness*
- Paradigma "preživijo najuspešnejši", *survival of the fittest*
- Funkcija uspeha je pravzaprav ciljna funkcija za optimizacijo

Drevesna predstavitev matematičnih izrazov

Primer predstavitve $y(x-3)-5$ z dvojiškim drevesom

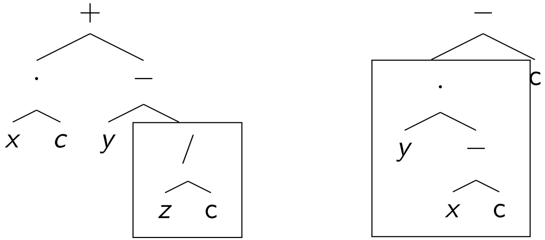


Vloga vozlišč v drevesu

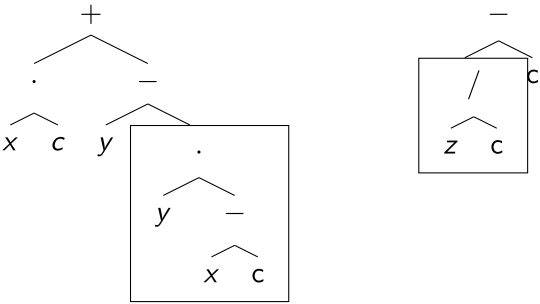
- Notranja vozlišča ustrezajo aritmetičnim operatorjem (ali funkcijam)
- Končna vozlišča ustrezajo spremenljivkam in konstantnim parametrom

Križanje dveh kromosomov: od staršev ...

- Starša $(x \cdot c) + (y - (z/c))$ in $(y \cdot (x - c)) - c$
- Pri vsakem staršu naključno izberemo notranje vozlišče
- Spodaj sta okvirjena poddrevesa od izbranih vozlišč



Križanje dveh kromosomov: . . . do potomcev



- Zamenjamo izbrani vozlišči (poddrevesi)
- Potomca $(x \cdot c) + (y - (y \cdot (x - c)))$ in $(z/c) - c$

Točkovna mutacija kromosoma

Izberemo naključno vozlišče

- Če je notranje, naključno zamenjamo aritmetični operator
- Če je končno, naključno zamenjamo
  - spremenljivko s konstanto, ali
  - spremenljivko z drugo naključno izbrano spremenljivko, ali
  - konstanto z naključno izbrano spremenljivko

Primera točkovne mutacije drevesa za izraz $(z/c) - c$

Splošna mutacija kromosoma

Izberemo naključno notranje vozlišče

In poddrevo zamenjamo z naključno ustvarjenim poddrevesom.

Primer splošne mutacije drevesa za izraz $(z/c) - c$

Ocenjevanje parametrov enačbe

Za podano enačbo $Y = E(X)$, npr. $d = \sqrt[3]{c \cdot p^2}$

- Upoštevamo množico podatkov S in poženemo optimizacijo

$$\boldsymbol{c}^* = \underset{c \in \mathbb{R}^m}{\arg\min} \, \| Y - E(X, \boldsymbol{c}) \|$$

- Tako dobimo $c^* = 7.51 \cdot 10^{-6}$ in torej enačbo

$$e = \left(d = \sqrt[3]{7.51 \cdot 10^{-6} \cdot p^2}\right)$$

Nato izračunamo napako enačbe

planet ID	$Y = d$	$X_1 = p$	$e(X)$	$(e(X) - Y)^2$
Merkur	0.389	87.77	0.386803488	4.82E-06
Venera	0.724	224.7	0.723871867	1.64E-08
Zemlja	1	365.25	1.000736637	5.43E-07
Mars	1.524	686.95	1.524788947	6.22E-07
Jupiter	5.2	4332.62	5.205071703	2.57E-05
Saturn	9.51	10759.2	9.54508141	1.23E-03
RMSE				1.45E-02

Napaka enačbe je torej $1.45 \cdot 10^{-2}$, uspešnost je $1/(1.45 \cdot 10^{-2})$.

Algoritem

function Evolutionary(S, $max.gen$, $pop.size$)
  $gen = 1$
  $pop =$Init($X, pop.size$)
  $pop =$Eval(pop)
  while $gen \leq max.gen$ **do**
    for $i = 1 \ldots pop.size$ **do**
      $new.pop = new.pop \cup$ ApplyOperator(pop)
    end for
    $pop =$Eval($new.pop$)
    $gen = gen + 1$
  end while
  return pop
end function

Pomožne funkcije

- Init: ustvari množico *pop.size* naključnih kromosomov (dreves)
- Eval: izračuna uspešnost kromosomov v podani učni množici
- ApplyOperator: naslednja prosojnica

Pomožna funkcija ApplyOperator

1. Naključno izbere enega izmed evolucijskih operatorjev
2. Za mutacijo iz populacije izbere en kromosom, za križanje dva
3. Nad izbranimi kromosomi izvede evolucijski operator

Izbira operatorja: parametri algoritma

- Verjetnost križanja
- Verjetnost(i) (različnih vrst) mutacije

Izbiranje kromosomov

Več možnosti

- Popolnoma naključna izbira enega izmed kromosomov
- Ruleta: verjetnost izbire kromosoma proporcionalna uspešnosti
- Turnirska izbira: izbor para, nato boljšega od dveh
- Elitna izbira: le med $p\%$ najboljših kromosomov

Upoštevanje kompleksnosti (zapletenosti) enačb

Mere kompleksnosti enačbe

- Število notranjih in/ali končnih vozlišč drevesa
- Število konstantnih parametrov
- Dolžina enačbe v karakterjih

Dve možnosti

1. Uspešnost izračunamo kot kombinacijo napake in kompleksnosti

$$1/(\mathit{Error} + \alpha \cdot \mathit{Complexity})$$

α je stopnja vpliva kompleksnosti: večji kot je, manj je preprileganja

2. Opazujemo oboje hkrati: več-ciljna optimizacija

Popularna implementacija

Eurequa (Schmidt in Lipson 2009)

`www.creativemachineslab.com/eureqa.html`

PySR@github (Cranmer 2020)

Nepreverjena Python implementacija, sloni na evolucijskem pristopu.

Propozicionalizacija

Vpeljava novih spremenljivk X_T z naborom transformacij podanih X

- Z množenjem do določene, omejene stopnje: X_1^2, X_1X_2, X_1X_3, ..., X_1X_p, X_2^2, X_2X_3, ..., X_2X_p, ..., X_p^2, ..., X_p^5
- Z apliciranjem funkcij, npr. trigonometričnih ali krožnih
- S kombinacijami enih in drugih transformacij

Rezultat: razširjen nabor spremenljivk $X_N = X \cup X_T$

Iskanje enačb v razširjenem naboru spremenljivk

Metoda Lagrange

- Naštevanje kombinacij (omejenega reda) spremenljivk iz X_N
- Linearna regresija za ocenjevanje parametrov

Metoda Sindy

- Redka linearna regresija v razširjenem naboru spremenljivk

$$\min_{\boldsymbol{\beta}} \sum_{(\boldsymbol{x}_N, y) \in S} (y - \boldsymbol{\beta}^T \boldsymbol{x}_N)^2 + \lambda \|\boldsymbol{\beta}\|_1$$

- Regularizacijski člen $\lambda \|\boldsymbol{\beta}\|_1$ poskrbi, da je malo parametrov $\beta \neq 0$
- λ je moč regularizacije: večji kot je, manj je preprileganja

Viri in implementacije

Lagrange (Džeroski in Todorovski 1993, 1995)

- Ni več delujoče implementacije
- Ali pač, `kt.ijs.si/ljupco_todorovski/ed/lg-www.tar.Z`

Sindy (Brunton in ost. 2016)

Python implementacija `pysindy@github`

Predznanje

Druga možnost za naslavljanje preprileganja.

Kako vpeljati predznanje v odkrivanje enačb?

Definicija prostora možnih enačb in gramatike

$$
\begin{array}{lcl}
E & \to & E + F \mid E - F \mid F \\
F & \to & F \cdot T \mid F / T \mid T \\
T & \to & const \mid V \mid (E) \\
\\
V & \to & X_1 \mid X_2 \mid \ldots \mid X_p
\end{array}
$$