Pēteris Račinskis
pr20015

Mājas darbs 1: ievads, asociācijas

Uzdevums 1

a. Iedzīvotāju vecuma spektra izrakšana

Dotais uzdevums - iedzīvotāju vecuma spektra konstruēšana, dalot noteikta platuma intervālos - ir vispārīgi formulējams kā histogrammas konstruēšana. Matemātiski katram intervālam atbilst vesels skaitlis, kas ir vienāds ar intervālam piederošo elementu skaitu. Konkrēti to iespējams definēt dažādi, bet kā piemēru varētu ņemt sekojošo:

$$m_{i,j} = |\{(x, y) \mid (x, y) \in M \wedge i \leq y - x < j\}|$$

kur M - visu dzimšanas un miršanas datumu pāru kopa (pēc nepieciešamajām filtrēšanas operācijām),

x - dzimšanas datums (pārveidots aprēķinam piemērotā reprezentācijā),

y - miršanas datums (pārveidots aprēķinam piemērotā reprezentācijā),

i - intervāla sākums,

j - intervāla beigas,

m - elementu skaits intervālā.

Algoritmiski šādu aprēķinu var triviāli veikt lineārā laikā pēc M izmēra. Sarežģītākā daļa no programmēšanas viedokļa būtu robusta aprēķinu veikšana ar datumiem, taču lielākā daļa augsta līmeņa programmēšanas valodu piedāvā bibliotēkas ar abstrakcijām, kas ļauj ar datuma datu tipiem veikt saskaitīšanas un atņemšanas operācijas kā ar veseliem skaitļiem.

b. Sagaidāmā dzīves ilguma modelēšana kā funkcija no pašreizējā vecuma gados

Vispārīgi šis uzdevums ir formulējams sekojoši:

$$l = E[x - x_0]$$

kur l - sagaidāmais dzīves ilgums,

x0 - pašreizējais vecums gados,

x - miršanas vecums.

Miršanas vecumu sadalījuma blīvumu iespējams aproksimēt ar kādu nepārtrauktu vai intervālos nepārtrauktu funkciju (piem., polinomu vai pa intervāliem kombinētu splainu) f(x), un tad varētu aprēķināt šo vērtību kā integrāli:

$$E[x - x_0] = \frac{1}{P[x > x_0]} \int_{x_0}^{\infty} f(x) x dx$$

kur 1/P[x > x0] - blīvuma funkcijas normalizēšana (lai f(x)dx integrālis definīcijas apgabalā būtu 1).

Ejot vienu soli tālāk, iespējams atrast funkcijas, kas pietiekami labi aproksimē paša integrāļa vērtības, un šādu pieeju arī izmanto praksē - piemēram, “Farr’s death rate method”, ko piedāvā dažādas sagaidāmā dzīves ilguma aprēķina pamācības[1].

Tā kā uzdevuma nosacījumos dotajā gadījumā dota datu kopa ar diskrētiem lielumiem, no kuras pašam jārok dati, droši vien vienkāršāk būtu šo uzdevumu atrisināt diskrētajā gadījumā. Pieņemot, ka jau atrisināts uzdevums, kurā jāsadala dati vienāda platuma *k* intervālos, rezultējošā histogramma ar intervālu skaitļiem m_ij ir izmantojamais modelis, **x ir dzīves ilgums (nevis miršanas vai dzimšanas datums!)** un pašreizējais vecums ir x0:

1. $k = const.; i_{max} = const.$
2. $x_i = i * k; i \in \{0, 1, \cdots, i_{max}\}$
3. $b = Max(\{i \mid x_i < x_0\})$
4. $k_{cor} = \frac{x_{b+1} - x_0}{k}$
5. $S = \sum_{i=b}^{i_{max}} m_{x_i, x_{i+1}} - \frac{1}{k_{cor}} m_{x_b, x_{b+1}}$
6. $x'_1 = Avg(x_0, x_{b+1}) - x_0$
7. $E[x - x_0] = \frac{1}{s}[k_{cor} x'_1 m_{x_b, x_{b+1}} + \sum_{i=b+1}^{i_{max}} m_{x_i, x_{i+1}} (Avg(x_i, x_{i+1}) - x_0)]$

Šis paņēmiens vienkārši saskaita intervālos esošo elementu skaitu virs x_0, taču nelielus sarežģījumus ievieš nepieciešamība pieskaitīt koriģēto intervālu, kurā ietilpst x_0. Tas tiek darīts, pieņemot, ka intervālā pastāv vienmērīgs sadalījums - iegūst konstanti k_cor, ar ko pareizina “tekošā” intervāla x_b - x_b+1 elementu skaitu. Kopējo elementu skaitu S iegūst, atņemot šī paša skaitļa reizinājumu ar inverso k_cor.

c. Kurus datus vajadzētu ignorēt?

Miršanas vecumu sadalījums būs mākslīgi zemāks, nekā īstais, ja tiks ņemti vērā nesen dzimušu jaunu cilvēku miršanas datumi - šo cilvēku vienaudži, kas dzīvos ilgāk, vēl neatrodas kapsētā. Tāpēc bez kaut kādām viltīgām korekcijām nevajadzētu izmantot pārāk nesen dzimušu cilvēku datus - vienkāršākais paņēmiens, kā tos atsijāt varētu būt ieviest kaut kādu pēdējo dzimšanas gadu, kas ir pietiekami sens, lai neviens vai gandrīz neviens šāda vecuma cilvēks vairs nebūtu dzīvs.

[1] https://ec.europa.eu/health/indicators/docs/echi_10_ds_en.pdf

Pēteris Račinskis
pr20015

Uzdevums 2

a. Datu kopas apraksts

Datu kopa ir dota .arff formātā, kur sākumā definēti katras datu rindas atribūti un to iespējamās vērtības, bet pēc tam ierakstītas pašas datu rindas.

Šajā gadījumā datu rinda ir “transakcija” - visticamāk pirkums - un atribūti ir dažādi parametri, kas ar šo transakciju ir saistīti. Lielākā daļa izskatās pēc produktiem, kas varētu būt pirkumā, taču ir arī citi, kas varētu būt, piemēram, veikala sadaļa (departmentxxx), no kuras paņemtās preces ir pirkumā. Vērtības lielākai daļai atribūtu ir definētas kā “t” (true) vai ? (undefined). Atribūts “total” var pieņemt vērtības “low” un “high”, bet “% low < 100”, visticamāk, ir komentārs (spriežot pēc .arff formāta dokumentācijas).

b. Asociāciju meklēšanas algoritma izvēle

Weka programmatūras pakotne piedāvā trīs iespējas, veicot asociāciju meklēšanu: apriori, fpgrowth un filtered associator. Trešais ir vienkārši iespēja, kas ļau izvēlēties kādu no pieejamajiem asociāciju algoritmiem un apvienot to ar kādu filtrēšanas algoritmu. Abos gadījumos iegūtie 10 labākie rezultāti ir tie paši, taču FpGrowth algoritms šķietami strādā ātrāk, taču datu kopa nav pietiekami liela, lai šī atšķirība būtu sevišķi jūtama. Tiešsaistē pieejamie resursi liek noprast, ka FpGrowth ir asimptotiski efektīvāks algoritms, jo būvē koka datu struktūru no datiem, kurus apstrādā[2].

c. Interesantas asociācijas

Visas “labākās” asociācijas, šķiet, ir saistītas ar “bread and cake” atribūtu. Arī izvēloties lielākus atrodamo nosacījumu skaitus (100. 1000) un mainot dažādus iestatījumus izdodas atrast tikai šādus noteikumus. Iespējams, ka var kaut kādā veidā likt programmatūrai ignorēt konkrētus rezultātus, taču autoram tas nav izdevies. Ja vien iegūtie rezultāti nav kāda īpatnēja programmatūras kļūme, izskatās, ka sakarības starp dazādiem ar saviesīgiem pasākumiem saistītiem produktiem un “bread and cake” absolūti dominē šajā datu kopā.

[2] FP Growth: Frequent Pattern Generation in Data Mining with Python Implementation