Algoritem BatMiner za rudarjenje asociativnih pravil

Iztok Fister ml. in Iztok Fister

→ **Razvoj spletnega računalništva dandanes spremljata dva med seboj tesno prepletena izziva: velika količina neraziskanih podatkov v podatkovnih bazah in eksponentna rast računske moči računalniških sistemov. Prvi izziv je pripeljal do nastanka moderne računalniške discipline podatkovno rudarjenje, katerega cilj je odkrivanje informacij, skritih v podatkih, medtem ko drugi izziv poskuša zadovoljiti vse večje zahteve spletnega računalništva po procesorski moči in velikosti pomnilniških medijev. Dejansko je prav zadnji omogočil veliko rast in razvoj podatkovnega rudarjenja v zadnjem desetletju.**

Podatkovno rudarjenje je multidisciplinarno področje, ki se zgleduje po principih ostalih znanstvenih področij, matematike, statistike, računalništva, fizike, inženirstva. Na to področje so imele največji vpliv naslednje discipline:

- statistika z uporabo statističnih metod in vizualizacijo podatkov,
- umetna inteligenca z uporabo metod strojnega učenja,
- metode računske inteligence in
- sistemi podatkovnih baz.

Dandanes se na tem področju pojavlja več vrst aplikacij, ki jih lahko razdelimo v napovedne in opisne. Prvi tip aplikacij je namenjen napovedovanju (npr. klasifikacija, regresija) vrednosti ene ali več spremenljivk v prihodnosti na podlagi dela spremenljivk v podatkovnih bazah, medtem ko se drugi tip (npr. gručenje, rudarjenje asociativnih pravil, odkrivanje zaporednih vzorcev) ukvarja z identifikacijo vzorcev za opis podatkov, shranjenih v podatkovnih bazah, in njihovo vizualizacijo na način, ki je enostavno razumljiv uporabnikom. V tem članku se osredotočamo na rudarjenje asociativnih pravil.

Rudarjenje asociativnih pravil je proces identificiranja pravil odvisnosti med objekti znotraj velikih transakcijskih podatkovnih baz [4]. S temi pravili iščemo povezave med objekti oziroma napovedujemo pojavitev objektov v primeru, da se pojavi določeno sosledje drugih objektov.

Formalna definicija rudarjenja asociativnih pravil je naslednja: Predpostavimo, da sta podani množica objektov $O = \{o_1, \ldots, o_n\}$ in množica transakcij T v transakcijski podatkovni bazi D, kjer je vsaka transakcija $t \in T$ podmnožica objektov $T \subseteq O$. Potem lahko asociativno pravilo definiramo kot implikacijo oblike

- $X \Rightarrow Y$, (1)

kjer velja $X \subset O$, $Y \subset O$ in $X \cap Y = \emptyset$. Množico mogočih pravil ocenimo z naslednjima meriloma [1]:

- $supp(X \Rightarrow Y) = \frac{|\{t \in T; X \cup t\}}{|T|}$ (2)

in

- $conf(X \Rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)}$, (3)

kjer podpora $supp(X \Rightarrow Y)$ označuje, kako pogosto se objekt X pojavlja v transakcijski podatkovni bazi in zaupanje $conf(X \Rightarrow Y)$, kako pogosto asociativno pravilo $X \Rightarrow Y$ vrača vrednost *pravilno*. Iz te množice izberemo tista pravila, ki izpolnjujejo naslednji relaciji:

- $supp(X \Rightarrow Y) \geq S_{min}$

in

- $conf(X \Rightarrow Y) \geq C_{min}$,

kjer S_{min} označuje minimalno zaupanje in C_{min} minimalno podporo. Do danes je bilo razvitih veliko algoritmov za rudarjenje asociativnih pravil, kot npr. Apriori, Eclat, FP-Growth.

Zadnjih nekaj let poskušajo raziskovalci reševati ta problem tudi z uporabo algoritmov po vzorih iz narave. Med algoritme po vzorih iz narave štejemo evolucijske algoritme in algoritme inteligence rojev. Oboji spadajo med populacijske algoritme, kar pomeni, da operirajo s populacijo rešitev. Prva vrsta posnema Darwinovo evolucijsko teorijo, po kateri imajo v naravi največ možnosti za preživetje najuspešnejši posamezniki. Druga vrsta pa temelji na obnašanju delcev znotraj roja delcev, kjer delci delujejo kot agenti, ki so sposobni izvajanja relativno enostavnih opravil. Če ti agenti delujejo povezani v skupnost, so sposobni izvajanja tudi kompleksnejših opravil.

Več informacij o teh algoritmih lahko najde bralec v članku [2].

Eden izmed algoritmov za rudarjenje asociativnih pravil je tudi BatMiner, ki ga predstavljamo podrobneje v nadaljevanju članka. Ta temelji na algoritmu na osnovi obnašanja netopirjev [5] in ga je za rudarjenje asociativnih pravil potrebno prilagoditi. Pri tem sta najpomembnejši dve:

- prilagoditev predstavitve rešitev, in
- prilagoditev ocenitvene funkcije.

Rešitev algoritma za rudarjenje asociativnih pravil BatMiner je predstavljena kot vektor realnih števil:

- $\mathbf{x}_i^{(t)} = (x_{i,1}^{(t)}, \ldots, x_{i,d}^{(t)}, x_{i,d+1}^{(t)}, x_{i,d+2}^{(t)})$,

kjer $x_{i,j}^{(t)} \in [0,1)$ za $i = 1, \ldots, n \wedge j = 1, \ldots, d$ kodira značilnice v asociativnem pravilu, $x_{i,d+1}^{(t)}$ označuje točko reza, $x_{i,d+2}^{(t)}$ pa smer asociativnega pravila. Spremenljivka n določa velikost populacije, d maksimalno število atributov v asociativnem pravilu in je t števec generacij. Točka reza določa, katere značilnice spadajo v predpostavko (angl. antecedent) in katere v posledico (angl. consequence) specifičnega asociativnega pravila.

Vsak element vektorja $x_{i,j}^{(t)}$ kodira dve vrsti informacije. Ko so elementi urejeni po naraščajočem vrstnem redu, pripadajoči indeksi tvorijo permutacijo značilnic, ki določa vrstni red pojavitve elementov v asociativnem pravilu. Povedano z drugimi besedami, glede na relacijo ›manjši ali enak‹ dobimo naslednjo relacijo urejenosti:

- $x_{i,\pi(i,1)}^{(t)} \leq x_{i,\pi(i,2)}^{(t)} \leq \ldots \leq x_{i,\pi(i,d)}^{(t)}$,

kjer $\pi(i,j)$ določa pripadajoči indeks atributa na j-ti poziciji i-tega vektorja.

Po drugi strani je območje dopustnih vrednosti značilnic v intervalu $x_{i,j}^{(t)} \in [0,1]$ za $j = 0, \ldots, d$ razdeljeno v $m_j + 1$ ekvidistantnih intervalov, kjer vsak inteval $[k, k+1]$ za $k = 0, \ldots, m_j$ ustreza enemu izmed elementov množice atributov j-te značilnice $o_{i,j} \in \{a_{i,0}, a_{i,1}, \ldots, a_{i,m_j}\}$ in parameter m_j označuje število elementov te množice. Atribut $a_{i,j}^{(t)}$ v generaciji t izračunamo po naslednji enačbi:

- $$a_{i,j}^{(t)} = \left\lfloor \frac{x_{i,j}^{(t)}}{m_j + 1} \right\rfloor, \quad \text{za } i = 0, \ldots, n \wedge j = 0, \ldots, d. \tag{4}$$

Atribut $a_{i,0}^{(t)} = \text{NULL}$ ima poseben pomen, saj določa, da pripadajoče značilnice ni v asociativnem pravilu.

Točko reza $p_i^{(t)}$ asociativnega pravila določa nadzorni parameter $x_{i,d+1}^{(t)}$ in jo dekodiramo po naslednji enačbi:

- $p_i^{(t)} = \lfloor x_{i,d+1}^{(t)}(d-2) \rfloor + 1, \quad \text{za } i = 0, \ldots, n,$

kjer dovoljujemo maksimalno $d-2$ točk reza v vsakem asociativnem pravilu.

Element $x_{i,d+2}^{(t)} \in [0,1]$ določa smer branja asociativnega pravila, ki ga dekodiramo po naslednji enačbi:

- $$q_i^{(t)} = \begin{cases} 0, & \text{če } x_{i,d+1}^{(t)} \leq 0.5, \\ 1, & \text{če } x_{i,d+1}^{(t)} > 0.5, \end{cases} \quad \text{za } i = 0, \ldots, n.$$

Značilnica	Atributi	Vrednosti
TRAJANJE	KRATKO	< 150 min
	SREDNJE	≥ 150 min $\wedge < 300$ min
	DOLGO	≥ 300 min
DOLŽINA	KRATKA	< 50 km
	SREDNJA	≥ 50 km $\wedge < 120$ km
	DOLGA	≥ 120 km
PORABA	MAJHNA	< 1200 kCal
	SREDNJA	≥ 1200 kCal $\wedge < 2800$ kCal
	VISOKA	≥ 2800 kCal
UTRIP	MAJHEN	< 130 BPM
	SREDNJI	≥ 130 BPM $\wedge < 170$ BPM
	VISOK	≥ 170 BPM

TABELA 1.
Diskretizacija zveznih spremenljivk, ki služijo kot značilnice.

TRAJANJE	DOLŽINA	PORABA	UTRIP	VREME	TIP	SPANJE	KRČI	p	q
KRATKO	KRATKA	∅	∅	∅	INTERVAL	∅	∅	3	0
Predpostavka			Posledica					Nadz. par.	

TABELA 2.
Primer veljavne rešitve.

Če je vrednost $q_i^{(t)} = 0$, asociativno pravilo beremo z leve proti desni, če je $q_i^{(t)} = 1$ pa z desne proti levi.

Ocenitvena funkcija v algoritmu BatMiner je podobna funkciji, uporabljeni v [3] in jo izrazimo na naslednji način:

- $f(\mathbf{x}_i^{(t)}) =$

$$\begin{cases} \frac{\alpha * conf(\mathbf{x}_i^{(t)}) + \gamma * supp(\mathbf{x}_i^{(t)})}{\alpha + \gamma}, & \text{če } feasible(\mathbf{x}_i^{(t)}) = true, \\ -1, & \text{drugače,} \end{cases}$$

kjer je $conf()$ merilo zaupanja, $supp()$ merilo podpore pravila, α in γ so uteži, namenjene uravnoteževanju vpliva zaupanja in podpore ter funkcija $feasible(\mathbf{x}_i)$, ki določa, če je rešitev dopustna ali ne. Naloga optimizacije je poiskati maksimalno vrednost ocenitvene funkcije.

Algoritem BatMiner uporabimo za ugotavljanje značilnostih športnika v športnem treningu. S športnimi aktivnostmi se namreč v današnjih časih začenja ukvarjati vse več ljudi, v kar jih največkrat prisili moderni življenjski slog. Ti športniki običajno spremljajo napredek svojega treniranja s pomočjo športnih ur oziroma mobilnih naprav, ki jih nosijo med treningom. Te naprave praviloma generirajo veliko število podatkov, ki lahko služijo športnim trenerjem pri načrtovanju športnih treningov, ugotavljanju trenutne pripravljenosti športnika v treningu, sestavljanju športnih jedilnikov ipd. V naši študiji uporabimo te podatke (tj. dolžino, trajanje, srčni utrip in porabo kalorij med treningom) kot osnovo za ugotavljanje značilnostih športnika v športnem treningu.

Pri tem podatke o spremljanju športnih treningov, pridobljenih z mobilnih naprav, dopolnimo s informacijami o psiho-fizičnem stanju športnika pred tre-

ningom (tj. vpliv vremena, tip treninga, nočno spanje pred treningom, morebitni krči) in vse skupaj shranimo v podatkovno bazo. Iz podatkov v podatkovni bazi izluščimo dejavnike, ki vplivajo na izvedbo športnega treninga posameznega športnika, in te shranimo kot značilnice (angl. features) v transakcijsko podatkovno bazo. Algoritem BatMiner za rudarjenje asociativnih pravil v tej bazi išče asociativna pravila, ki so za športnega trenerja lahko zelo uporabna pri napovedovanju športnikove forme ali odkrivanju problemov, povezanih s športnim treningom oziroma tekmovanji.

V našem primeru imamo opravka z osmimi značilnicami predstavljenimi kot zvezne oziroma diskretne spremenljivke. Zvezne značilnice, dobljene iz mobilnih naprav, je potrebno najprej diskretizirati. Primer diskretizacije podatkov, pridobljenih z mobilnih naprav, je prikazan v tabeli 1. Omenjena diskretizacija je narejena na osnovi teorije športnega treninga in velja tako za profesionalne kot amaterske športnike.

Diskretne značilnice, ki označujejo psiho-fizično stanje športnika, imajo preddefinirano število atributov. V našem primeru so to:

- VREME = {SONČNO, OBLAČNO, DEŽEVNO, SNEŽENO},
  TIP = {RAZPELJAVA, INTERVAL, MOČ, VZDRŽLJIVOST},
  SPANJE = {DOBRO, SREDNJE, SLABO},
  KRČI = {BREZ, RAHLI, VELIKI}.

Primer predstavitve asociativnega pravila, ki ga je odkril algoritem BatMiner v traksakcijski podatkovni bazi z 80 transakcijami, prikazuje tabela 2, kjer nadzorni parameter $p = 3$ pomeni točko reza, ki deli pravilo na predpostavko in posledico, in kjer nadzorni parameter $q = 0$ določa smer branja asociacijskega pravila z leve proti desni. Če predpostavimo, da značilnico združimo z atributom s pomočjo operacije združevanja (znak ›_‹), posledično iz rešitve dekodiramo naslednje asociativno pravilo

- TRAJANJE_KRATKO ∧ DOLŽINA_KRATKA ⇒ TIP_INTERVAL,

ki pravi: Če je trening kratke dolžine in kratkega trajanja, gre za intervalni tip treninga. Seveda je pravilo v skladu s teorijo športnega treninga, saj gre pri intervalnih treningih za zelo intenzivne kratkotrajne treninge kratkih dolžin.

Kot prikazuje zgornji primer, so algoritmi po vzorih iz narave uporabni tudi pri rudarjenju asociativnih pravil. V današnji družbi se ne moremo izogniti veliki rasti podatkov, ki nastajajo praktično na vsakem koraku, lahko pa se iz njih veliko novega naučimo. V prihodnosti lahko pričakujemo, da se bodo podobne rešitve z algoritmi po vzorih iz narave za podatkovno rudarjenje začele uporabljati tudi na ostalih področjih človekove dejavnosti.

Literatura

[1] R. Agrawal, T. Imielínski in A. Swami, *Mining association rules between sets of items in large databases*, ACM SIGMOD Record, 22(2), 207–216, 1993.

[2] I. Fister Jr., X.-S. Yang, I. Fister, J. Brest in D. Fister, *A brief review of nature-inspired algorithms for optimization*, Elektrotehniški vestnik, 80(3), 116–122, 2013.

[3] K. E. Heraguemi, N. Kamel in H. Drias, *Association rule mining based on bat algorithm*, In Bio-Inspired Computing-Theories and Applications, 182–186, Springer, 2014.

[4] G. Hrovat, G. Stiglic, P. Kokol in M. Ojsteršek, *Contrasting temporal trend discovery for large healthcare databases*, Computer methods and programs in biomedicine, 113(1), 251–257, 2014.

[5] K. Ljubič in I. Fister Jr., *Algoritem na osnovi obnašanja netopirjev*, Presek, 42(3), 26–28, 2015.

× × ×