

# Evolucija drveta klasifikacije upotrebom genetičkog programiranja

*Petar Zečević 169/2016 i Aleksandra Ružić 47/2016*

# Uvod

Drvo klasifikacije predstavlja model koji podacima iz skupa podataka sa određenim atributima, treba da odredi klase tj. vrednosti ciljne promenljive. Model (drvo) se pravi nad trening skupom, a testira nad test skupom (u oba su poznate vrednosti ciljne promenljive). Drvo klasifikacije je stablo u čijim su unutrašnjim čvorovima uslovi, a u listovima klase ciljne promenljive. Uslovi su oblika 'atribut' 'relacija' 'vrednost', gde je 'atribut' ime kolone iz skupa podataka, 'relacija' je operacija iz dozvoljenog skupa operacija (što je {<, <=, >, >=, ==, !=} ako je atribut numerički, u suprotnom {==, !=}), a 'vrednost' je moguća vrednost iz dozvoljenog skupa vrednosti tog atributa. Kada se primer iz skupa podataka "provlači" kroz drvo, za svaki čvor se proverava da li podatak ispunjava uslov. Ako ispunjava, ide u levi, a u suprotnom u desni čvor. List do kog dođe, određuje njegovu vrednost ciljne promenljive.

Nalaženje optimalnog drveta klasifikacije genetskim programiranjem podrazumeva da su stabla jedinke i da se operatori genetskog algoritma (selekcija, mutacija, ukrštanje) implementiraju u skladu s tim. U radu će biti razmatran blokovski pristup genetskom algoritmu koji traži drvo klasifikacije. Dodatno, zbog specifičnosti problema i kompleksnosti jedinke, uvode se dodatni operatori (potkresivanje, mutacija relacije, mutacija vrednosti, dodavanje gradivnih blokova).

Rad opisuje implementaciju varijante BGP algoritma (Building block approach to Genetic Programming). BGP je evolutivni algoritam koji služi za formiranje stabla odlučivanja. Ideja BGP algoritma je da početna populacija ima jedinke što manje dubine i da se kroz generacije ta dubina kontrolisano povećava. Time se obezbeđuje da, za rezultat, algoritam bira stablo sa što manjom dubinom, a što znači da se izbegavaju preprilagođeni modeli i modeli koji u podstablama sadrže međusobno kontradiktorna ili suvišna pravila. Rezultati i implementacija algoritma će biti poređeni sa rezultatima i idejnom implementacijom BGP-a u radu "Searching the Forest: Using Decision Trees as Building Blocks for Evolutionary Search in Classification Databases" autora SE Rouwhorst i AP Engelbrecht [1]. Algoritam je implementiran u programskom jeziku Python.

## Opis BGP algoritma

Jedinke algoritma su stabla. Inicijalna populacija se formira od nasumičnih stabala dubine 1 (dakle, koren u kom je uslov i dva lista u kom su klase). Veličina populacije se bira eksperimentalno. Ocena kvaliteta jedinke je preciznost (mogu se koristiti i druge ocene kvaliteta stabla). Operatori i njihovi opisi su sledeći:

- *mutacija relacije*: bira se nasumično unutrašnji čvor stabla, i bira se nasumična relacija iz skupa dozvoljenih relacija za taj atribut i postavlja se kao relacija u uslovu u tom čvoru.
- *mutacija vrednosti*: bira se nasumično unutrašnji čvor stabla, i bira se nasumična vrednost iz skupa dozvoljenih vrednosti za taj atribut i postavlja se kao vrednost u uslovu u tom čvoru.
- *ukrštanje*: za obe jedinke bira se nasumično unutrašnji čvor, i podstabla ispod datih čvorova se zamenjuju. Izabrani čvorovi ne moraju biti na istoj dubini.

- *potkresivanje*: bira se nasumično unutrašnji čvor stabla i zamenjuje se listom. Potkresivanje se vrši da bi se izbeglo preprilagođavanje.
- *selekcija*: turnirska. Ako je veličina turnira  $k$ , bira se iz populacije nasumičnih  $k$  jedinki i one "učestvuju u turniru". Pobjednik turnira učestvuje u ukrštanju. U jednoj varijanti ove selekcije, pobjednik je jedinka sa najvećom prilagođenošću. U drugoj, izabrane turnirske jedinke se sortiraju po prilagođenosti, i šansa da pobedi  $i$ -ta jedinka je  $p \cdot (1-p)^{(i-1)}$ , gde je  $p$  korisnički definisan parametar koji se (pogodili ste) eksperimentalno određuje. Postoje i mnoge druge varijante ove selekcije [2]. Veličina turnira se bira eksperimentalno.
- *dodavanje gradivnih blokova*: bira se nasumično list iz stabla i zamenjuje se nasumičnim drvetom dubine 1.

Za svaki od operatora, osim selekcije, postoji po parametar koji predstavlja verovatnoću njegovog dešavanja. Ti parametri se određuju eksperimentalno. Takođe, da bi se dodavao gradivni blok na jedinke, potrebno je da bude ispunjen sledeći uslov:  $(avg\_depth + avg\_width) - (prev\_avg\_depth + prev\_avg\_width) < L$ , gde su:

- $avg\_depth$  - prosečna dubina stabla u trenutnoj generaciji
- $avg\_width$  - prosečna širina stabla u trenutnoj generaciji (broj listova)
- $prev\_avg\_depth$  - prosečna dubina stabla u prošloj generaciji
- $prev\_avg\_width$  - prosečna širina stabla u prošloj generaciji
- $L$  - korisnički zadat parametar. Određuje se eksperimentalno.

Kriterijum zaustavljanja je  $min\_rule\_acc < e^{(c * trainsize * A)}$ , gde su:

- $A = 1/T(0) - 1/T(t)$ , gde je  $T(t) = T(0) - t$  "temperaturna" funkcija broja iteracija (sa povećanjem broja iteracija, funkcija "hladi" program, smanjujući gornji izraz, a time povećavajući mu šansu da se zaustavi).  $T(0)$  je korisnički zadat parametar. Određuje se eksperimentalno.
- $trainsize$  - veličina trening skupa
- $min\_rule\_acc$  - najmanja preciznost među pravilima
- $c$  - korisnički zadat parametar. Određuje se eksperimentalno.

Algoritam počinje od formiranja inicijalne populacije. Onda u se svakoj generaciji, redom, ispituje da li mogu da se dodaju gradivni blokovi na jedinke iz populacije, biraju jedinke za ukrštanje, vrši ukrštanje, mutacije i potkresivanje. Pamti se i ažurira u svakoj generaciji najbolja jedinka. Generacije se smenjuju dok se ne ispuni kriterijum zaustavljanja.

## Implementacija i razlike

U radu, drvo je direktno implementirano kao stablo. Čvor je predstavljen klasom koja ima dve potklase: jedna za unutrašnje čvorove i jedna za listove. Klasa za unutrašnje čvorove sadrži atribut koji označavaju levog potomka, desnog potomka i uslove čvora (atribut, relacija i vrednost). Klasa za listove sadrži atribut koji označava klasu. Klasa za čvor ima i atribut indeks koji označava poziciju čvora u drvetu, a pozicije se dodeljuju kao za balansirano binarno drvo. Balansirano drvo je binarno stablo kod kog za svaki čvor važi da je apsolutna vrednost razlike dubina njegovih podstabala najviše 1. Indeksi kod takvih stabala se dodeljuju na sledeći način: koren je indeksa 1, a deca čvora sa indeksom  $i$ , imaju indekse  $2i$  i  $2i+1$ . Klasa koja predstavlja drvo sadrži atribut koji predstavlja koreni čvor. Drvo se instancira tako što se napravi koren, a zatim se dodaju čvorovi na mesta njegovih potomaka. Takođe, zato što drvo nije balansirano, klasa za drvo takođe ima atribut koji označava

skup zauzetih indeksa, odnosno pozicija, da bi se omogućio pristup bilo kom čvoru iz stabla (što je neophodno da bismo mogli pristupiti nasumičnom čvoru za bilo koji od gore navedenih operatora). Klasa za drvo ima metod koji izračunava preciznost drveta.

Umesto datog izraza za zaustavljanje, za kriterijum zaustavljanja smo koristili maksimalan broj iteracija. Napravljane su tri verzije funkcije koje se razlikuju u kriterijumu za dodavanje gradivnih blokova. Prva nema kriterijum za dodavanje gradivnih blokova. Druga verzija koristi kriterijum iz gore navedenog rada. Treća verzija koristi sledeći uslov za dodavanje gradivnog bloka: `avg_depth >= tree_depth AND avg_width >= tree_width` gde su:

- `avg_depth`- prosečna dubina stabla u trenutnoj generaciji.
- `tree_depth`- dubina stabla koje se razmatra za nadogradnju.
- `avg_width`- prosečna širina stabla (broj listova) u trenutnoj generaciji.
- `tree_width`- širina stabla (broj listova) koje se razmatra za nadogradnju.

Rezultati sve tri verzije će biti upoređivani, međusobno i sa rezultatom iz referisanog rada.

## Literatura:

1. SE Rouwhorst, AP Engelbrecht: "Searching the Forest: Using Decision Trees as Building Blocks for Evolutionary Search in Classification Databases", 2000.
2. Predrag Jančić, Mladen Nikolić: "Veštačka inteligencija", Beograd, 2019.