Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 004.9
DOI: https://doi.org/10.24867/19BE10Kovacevic

ANALIZA I POREĐENJE PERFORMANSI K-D STABLA I LOPTASTOG STABLA KAO PROSTORNIH STRUKTURA PODATAKA U DVODIMENZIONALNOM PROSTORU

ANALIYSIS AND COMPARISON OF K-D TREE AND BALL TREE PERFORMANCES AS SPATIAL DATA STRUCTURES IN TWO-DIMENSIONAL SPACE

Milena Kovačević, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj – *Sa konstantnim povećanjem skupova podataka, proces obrade i pristupa konkretnim podacima se komplikuje. Poseban slučaj predstavljaju podaci koji zahtevaju obradu u višedimenzionalnom prostoru. U ovom radu su prikazane dve prostorne strukture podataka koje služe za skladištenje, efikasan pristup i obradu velikih skupova podataka u više dimenzija. Predstavljene su napredne strukture podataka: k-d stablo i loptasto stablo, analizirane su njihove osobine i performanse i prikazani su rezultati o uspešnosti jedne u odnosu na drugu.*

Ključne reči: *binarno stablo, k-d stablo, loptasto stablo, metod k-najbližih suseda*

Abstract – *Because of the constant increase in the amount of data sets, processing and accessing specific data in large data sets becomes more complicated. A special case are data that require processing in multidimensional space. This paper presents two spatial data structures that can be used for storage, efficient access and processing of large data sets in multiple dimensions. In this paper k-d tree and ball tree, advanced data structures, with their properties are presented. Also, their performances have been analyzed and the results of the comparision have been shown.*

Keywords: *binary tree, k-d tree, ball tree, k-nearest neighbors search*

1. UVOD

Osnovne strukture podataka kao što su: liste, nizovi, stekovi, redovi, rečnici i druge, inicijalno su bile implementirane za manipulaciju i pretragu podataka koji se nalaze u jednoj dimenziji. Način smeštanja i organizovanja podataka je fundamentalna stvar u programiranju. Osnova većine programa ne leži u načinu obrade podataka, već u načinu na koji su podaci smešteni, organizovani i koliko im je lako i brzo moguće pristupiti. Strukture podataka pružaju jasniju sliku korisniku o podacima i njihovom međusobnom odnosu zbog čega je izbor odgovarajuće strukture podataka veoma važan za efikasan rad celog sistema.

Ustaljen način za obradu i pristup podacima bio je preko jedinstvenog identifikatora, koji omogućava pristup podacima prvenstveno u jednodimenzionalnom prostoru.

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bila dr Dunja Vrbaški, docent.

Razvojem računarske industrije nastaju i velike količine podataka i istovremeno se javlja problem smeštanja i obrade istih. Zbog pomenutih problema dolazi do potrebe za novim strukturama podataka kao i za drugačijim načinom čuvanja i posmatranja podataka. Sa jedinstvenim identifikatorom je bilo moguće predstaviti i pristupati podacima i u više dimenzija, ali na manje prirodan način i sa značajnim smanjenjem performansi celog sistema. Konstrukcija pogodne strukture za pronalazak određenog broja najbližih suseda jednog entiteta iz skupa podataka kao i sam proces nalaženja određenog broja najbližih suseda za dati entitet su problemi koji se javljau prilikom pretrage multimedijalih informacija, rudarenja podataka (eng. *data-mining*) i mašinskog učenja (eng. *machine learning*). Istovremeno, za sve ove probleme su potrebna sve efikasnija rešenja s obzirom da skupodvi podaka iz dana u dan postaju masovniji.

Cilj ovog rada je upoznavanje sa dve prostorne strukture podataka koje omogućavaju skladištenje i efikasan pristup podacima u više dimenzija, analiza njihovih osobina i poređenje njihovih razlika prilikom obrade različitih količina podataka u dve dimenzije. Za postizanje ovog cilja implementirani su k-d stablo i loptasto stablo i algoritam za efikasan pronalazak određenog broja najbližih suseda, a zatim su vršene analize performansi i poređenja nad pomenutim stablima.

Preostali deo rada organizovan je na sledeci način. U naredom poglavlju će biti prikazani radovi koji se bave sličnom problematikom. U trećem poglavlju će biti objašnjen način rada i konstrukcija prostornih struktura podataka: k-d stabla i loptastog stabla. U četvrtom poglavlju će biti reči o problemu pronalaska k-najbližih suseda i načinu implementacije ovog algoritma kod k-d stabla i loptastog stabla. U petom poglavlju će biti predsavljene performanse i razlike jednog stabla u odnosu na drugo, dok će u šestom poglavlju biti izložen zaključak i navedeni pravci u kojima bi se rad mogao razvijati i nadograđivati.

2. PRETHODNA REŠENJA

Brojne napredne strukture podataka su predstavljene u literaturi, uključujući k-d stablo [1], *quadtree* [2], loptasto stablo [3], pokrivajuće stablo [4] i r-stablo [5]. Među njima, k-d stablo je efikasna struktura podataka koju je izložio Bentli [1] 1975. godine. Jednostavna je, a može efikasno da obrađuje mnoge vrste upita. Bentli je pokazao da k-d stablo ima efikasnost $O(\log n)$ za upis novog podatka i brisanje postojećeg podatka iz stabla. K-d stablo [6] je jedno od

stabala za particionisanje prostora i organizovanje podataka u višedimenzionalnom prosotru. Kompleknost izgradnje ovog stabla od n tačaka je $O(n \log n)$

Loptasto stablo [3] je binarno stablo koje služi za održavanje hijerarhijskog uređenja među podacima kao i kod k-d stabla. Svaki čvor u loptastom stablu predstavlja loptu koja sadrži skup tačaka ograničenih hipersferom. Ukoliko nisu listovi, svaki čvor sadrži jednog ili dva naslednika koji zajedno sadrže sve tačke koje se nalaze u tom čvoru. Kao i k-d stablo, loptasto stablo se takođe može kreirati *top-down* algoritmom. Kompleksnost kreiranja loptastog stabla za n tačaka je $O(n \log n)$.

3. OPIS I IMPLEMENTACIJA

Stabla koja će biti objašnjena u ovom poglavlju su prostorne strukture podataka koje su diznajnirane za smeštanje, pretragu i rad sa podacima u više dimenzija. Dve osnovne ideje prostornih struktura podataka su: eksplicitno indeksiranje određenog podatka i uslov da sortiranje u strukturi podataka istovremeno izaziva i particionisanje prostora.

3.1. K-d Stabla

K-d stabla služe za sladištenje konačnog skupa podataka u k-dimenzionalnom prostoru, gde k predstavlja broj dimenzije. Ova stabla su modifikacija binarnog stabla pretrage. Iako se k-d stabla mogu primeniti za probleme u bilo kojoj dimeneziji, najčešće se koriste za rešavanje problema u dve ili tri dimenzije. Svaki unutrašnji čvor stabla predstavlja hiperravan koja seče prostor na dva dela, odnosno deli prostor po nekoj od dimenzija. U dvodimenzionalnom prostoru, čvor predstavlja liniju, dok u tri dimenzije predstavlja ravan. Prilikom prikaza, svaki čvor predstavlja jednu tačku u prostoru.

3.1.1. Konstrukcija k-d stabla

Za konstrukciju k-d stabla izabran je kanonički metod, kako zbog njegove efikasnosti, tako i zbog lakšeg poređenja performansi sa loptastim stablom. K-d stablo se sastoji od čvorova od kojih svaki čvor čuva infomaciju o jednoj pojavi iz skupa podataka, dimenziju po kojoj će deliti prostor, referencu na levo podstablo i referencu na desno podstablo. Konstrukcijom k-d stabla nastaće binarno stablo gde će svaki naredni red deliti prostor po nekoj dimenziji i gde se dimenzije kružno smenjuju. Prilikom kreiranja stabla nije neophodno poznavati ceo skup podataka. Konstrukcija k-d stabla kanoničkom metodom se vrši prolaskom kroz sledeće faze:

1. Određivanje korena stabla u prvoj iteraciji, odnosno korena podstabala u ostalim iteracijama procesa. Za koren stabla se bira medijana vrednosti po određenoj dimenziji. Medijana se pronalazi sortiranjem elemenata po najrasprostranjenijoj dimenziji i uzimanjem srednjeg elementa iz sortiranog niza elemenata.
2. Raspoređivanje ostalih pojava u levo podstablo u slučaju da se nalaze sa leve strane korena, odnosno raspoređivanje pojava u desno podstablo u slučaju da se nalaze sa desne strane korena stabla.
3. Rekurzivno ponavljanje u nastalim podstablima dok se svi elementi iz skupa podataka ne smeste u neki čvor ovog stabla.

3.2. Loptasto stablo

Lopstasto stablo je hijerarhijsko binarno stablo koje služi za skladištenje podataka u višedimenzionalnom porsotru. Sam naziv nastao je zbog načina na koji se vrši podela prostora. Za razliku od k-d stabla, kod koga se prostor deli po određenoj osi, kod loptastog stabla prostor se deli pomoću kružnica, tako da se svi podaci nalaze u unutar kružnice. Ova stabla dele prostor u niz hipersfera tako da podaci koji se nalaze blizu čine jednu sferu. Korenski čvor stabla je čvor koji zauzima najveći postor i obuhvata sve elemente skupa podataka. Svi ostali čvorovi su potomci korenskog čvora i razlikujemo čvorove koji su listovi od onih koji to nisu. Čvor koji je list sadrži skup tačaka koje on predstavlja i nema levo niti desno podstablo. Broj elemenata koji se mogu naći u listu treba biti prethodno definisan. Čvor koji nije list, pored toga što sadrži tačke koje se u njemu nalaze, ima reference na dva podstabla za koja važi:

1. Presek levog i desnog podstabla je prazan skup. Ova dva skupa mogu da imaju zajedničku površinu, ali ne i elemente.
2. Unija ovih skupova obuhvata sve podatke koji se nalaze u roditeljskom čvoru.

Podela prostora kod ovog stabla zavisi od implementacije i rasporeda tačaka. Krećući se od korena ka listovima površina kruga se gotovo duplo smanjuje na svakom narednom nivou.

3.2.1. Konstrukcija loptastog stabla

Konstrukcija loptastog stabla rađena je po k-d algoritmu. U ovoj implementaciji, svaki čvor sadrži određen broj podataka. Prilikom implementacije neophodno je poznavati ceo skup podataka koji će se smestiti u stablo.
Prvi korak u kreiranju loptastog stabla je određivanje korena, odnosno centra i poluprečnika kruga koji obuhvata sve tačke iz skupa tačaka. Proces određivanja ovog čvora prolazi kroz sledeće faze:

1. Slučajnim izborom se bira jedna tačka iz skupa.
2. Nalazi se tačka koja je najudaljenija od prethodno slučajno izabrane tačke.
3. Traži se druga tačka koja je najudaljenija od tačke koja je izabrana u prethodnom koraku. Tačke koje su izabrane u koraku 2. i 3. će biti najudaljenije tačke u celom početkom skupu tačaka.
4. Kroz dve pomenute tačke se provlači prava i vrši se projekcija svih ostalih tačaka na tu pravu.
5. Pomoću projektovanih tačaka na pravu, određuje se medijana. Medijana će biti centar korena, a poluprečnik korena će biti udaljenost od medijane do najdalje tačke iz skupa tačaka.

Drugi korak u procesu kreiranja loptastog stabla je rekurzivno deljenje krugova na po dva kruga, koja će zajedno obuhvatati sve elemente koje njihov roditelj sadrži. Ovaj proces deljenja krugova na dva manja se odvija dok god ne nastanu dovoljno mali krugovi koji će obuhvatati dozvoljen broj entiteta iz skupa.

4. ALGORITAM ZA PRETRAGU K-NAJBLIŽIH SUSEDA (KNNS)

Pronalazak najbližeg suseda (eng. *nearest neighbor search (NNS)*) je optimizacioni problem čiji je zadatak da se pronađe tačka u datom skupu podataka koja je najbliža, ili po nekom kriterijumu najsličnija, datoj tački. Najčešće korišćen način za određivanje bliskosti je računanje udaljenosti pomoću euklidskog, menhetn ili nekog drugog algoritma za računanje udaljenosti.

Pronalazak k-najbližih suseda je složenija verzija *NNS* algoritma gde je, umesto pronalaska jednog suseda, neophodno pronaći k suseda, gde k predstavlja broj najbližih suseda. Složenost pronalaska k suseda postaje posebno izražena sa povećanjem skupa podataka u kome se pretraga vrši i prilikom povećanja dimenzionalnosti prostora u kome se pretraga izvršava.

4.1. Konstrukcija KNNS algoritma za k-d stablo

Na samom početku pretrage neophodno je imati informacije o korenu k-d stabla, tačku T za koju tražimo određen broj najbližih suseda i k, odnosno broj najližih suseda koji je neophodno pronaći.

Za potrebe ovog rada pretraga je implementirana pomoću dve metode gde prva metoda služi za proveru prosleđenih parametara i za poziv druge metode u kojoj se nalazi glavna logika za pomenuti algoritam. U nastavku je dat opis implementiranih koraka:

1. Funkcija se izvršava rekurzivno i na početku svakog izvršavanja se prvo proverava da li je prosleđeni korenski čvor zadovoljavajuć i u slučaju da nije, završava se trenutna iteracija.
2. Nakon provere se određuje udaljenost između prosleđene tačke i centra trenutnog čvora.
   a. Ukoliko čvor već ne postoji u listi najbližih suseda i lista ima manje od k elemenata, čvor se dodaje u listu najbližih suseda.
   b. Ukoliko čvor ne opostoji u listi najbližih suseda i ako je bliži tački T od najdaljeg čvora iz liste najbližih suseda, vrši se zamena ta dva čvora, odnosno dodaje se bliži čvor u listu i izbacuje se najdalji čvor iz liste.
3. Vrši se sortiranje liste kako bismo lako mogli da pristupimo najbližem, ali i najudaljenijem elementu iz liste.
4. Određuje se trenutna dimenzija po kojoj se vrši podela prostora.
5. Vrši se provera da li je potrebno nastavljati pretragu. Pretragu je potrebno nastaviti ukoliko je trenutni čvor bliži tački T nego što je to najdalji element iz liste ili ako lista najbližih suseda nije popunjena.
6. Ukoliko se trenutni čvor nalazi sa leve strane tačke T, rekurzivno se poziva ova funkcija nad levim, pa zatim nad desnim podstablom. Ukoliko je slučaj obrnut, vrši se rekurzivno pozivanje metode prvo nad desnim, pa onda nad levim podstablom.
7. Algoritam se završava ili kada prosleđeno postablo ne postoji ili kada uslov za nastavak pomenut u koraku 5. nije ispunjen.

4.2. Konstrukcija KNNS algoritma za loptastno stablo

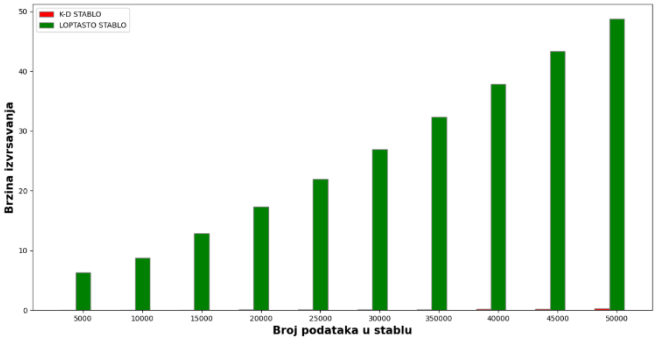
Konstrukcija KNNS algoritma za loptasto stablo je urađena na sličajn način kao kod k-d stabla. Ovde takođe postoje dve metode gde jedna vrši proveru prosleđenih parametara, a u drugoj se nalazi rekurzivni kod. Pored ove dve metode postoji i metoda za dodavanje čvora u listu k najbližih suseda ako su svi uslovi zadovoljeni. Algoritam za implementaciju u glavnoj metodi prolazi kroz naredne korake:

1. Provera da li je prosleđen koren stabla postojeći
2. Ukoliko je koren u trenutnoj iteraciji istovremeno i list poziva se metoda za dodavanje elementa u listu k suseda.
3. Čvor biva dodat u listu suseda ukoliko u njoj već ne postoji, ako ona nije potpuno popunjena ili ukoliko je bliži tački T od čvora iz liste koji je najudaljeniji od tačke T. U poslednjem slučaju najudaljeniji čvor biva izbačen iz liste k najbližih suseda.
4. Vrši se provera da li je potrebno nastavljati pretragu.
5. Ukoliko je uslov iz koraka 4. ispunjen, rekuzivno se vrši pozivanje pretrage u levom i desnom podstablu loptastog stabla.
6. Algoritam se završava kada prosleđeno stablo ne postoji, što se dešava kada se dođe do lista stabla ili kada nisu zadovoljeni uslovi da se rekrzivno pozivanje nastavi.

Ovakva pretraga u kombinaciji sa pomenutim strukturama postaje veoma efikasna, pošto se zbog načina na koji su elementi organizovani smanjuje broj provera i poređenja značajnog elemenata sa traženim elementom.

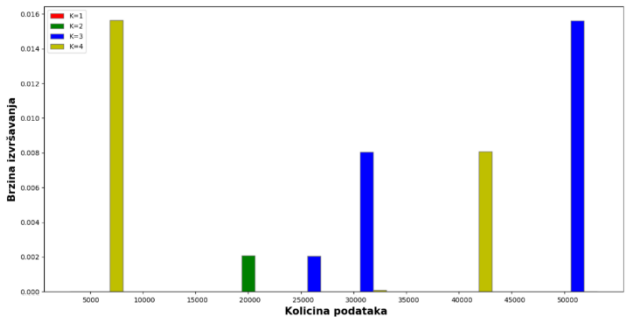
5. PRIKAZ REZULTATA I DISKUSIJA

U ovom poglavlju će biti predstavljeni rezultati poređenja performansi konstrukcije k-d stabla i loptastog stabla za različit broj podataka, kao i performansi nalaženja k najbližih suseda.



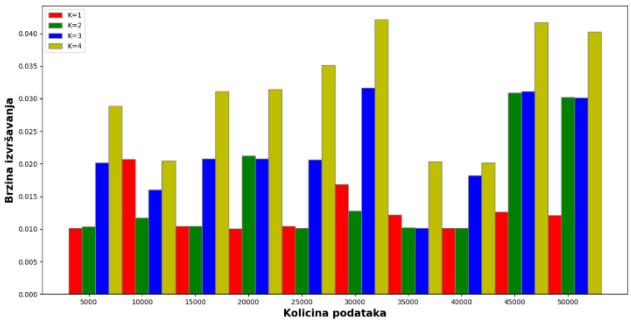
Slika 1. *Razlika u brzini konstrukcije k-d sabla i loptastog stabla*

Na slici 1. prikazano je vreme koje je neophodno za konstrukciju k-d stabla i loptastog stabla za različit broj elemenata. Broj elemenata skupa koji se koristi za konstrukciju prikazan je na x osi i kreće se od 5 000 do 50 000. Zeleni stubovi prestavljaju brzinu konstrukcije loptastog stabla za različite količine podataka, dok crveni stubovi, koji se tek naziru uz x osu, prikazuju brzinu konstrukcije k-d stabla. Na osnovu ovog grafika jasno se može utvrditi da je konstrukcija k-d stabla značajno brža od konstrukcije loptastog stabla.



Slika 2. *Vreme nalaska k najbližih suseda kod k-d stabla*

Na slici 2. je prikazano vreme koje je potrebno da se nađe 1, 3, 5 ili 10 najbližih suseda mežu deset različitih skupova podataka, gde najmanji skup podataka sadrži 5 000 podataka, a najveći 50 000 podataka. Sa povećanjem broja suseda koji se trebaju naći povećava se i prosečno vreme potrebno za pretragu.



Slika 3. *Vreme nalaska k najbližih suseda kod loptastog stabla*

Na slici 3. prikazano je vreme za pronalazak k najbližih suseda kod loptastog stabla. Treba napomenuti da ne samo da je pronalazak k najbližih suseda vršen nad istim podacima kao i kod k-d stabla već su susedi traženi za iste tačke. I na ovom grafiku možemo uočiti povećanje potrebnog vremena za nalazak k najbližih suseda prilikom povećanja broja najbližih suseda koje treba naći, ali je prosečna brzina potrebna da pronalaženje bilo kog od prikazanih brojeva suseda značajno veća kod loptastog stabla u odnosu na k-d stablo.

6. ZAKLJUČAK

U ovom radu su predstavljena dve prostorne strukture podataka za čuvanje i manipulaciju podacima u višedimenzionalnom prostoru: k-d stablo i loptasto stablo. Oba stabla su implementirana na sličan način kako bi se preciznije mogla odrediti razlika u njihovim performansama. Izvršena su merenja brzine konstrukcije ovih stabala i pretrage k najbližh suseda za određenu tačku u dvodimenzionalnom prosotru. Na osnovu izvršenih merenja pokazano je da su performanse konstrukcije k-d stabla značajno bolje od performansi konstrukcije loptastog stabla. Prilikom pretrage k najbližih suseda kod oba stabla se može uočiti smanjenje performansi sa porastom broja k, ali su performanse k-d stabla i u ovom domenu značajno bolje od performansi loptastog stabla.

Jedan od pravaca u kojima bi ovaj rad mogao da se unapredi jeste određivanje ponašanja i performansi k-d stabla i loptastog stabla u većim dimenzijama kao i uvođenje paralelizma prilikom konstrukcije stabala i prilikom pronalaska k najbližih suseda za određeni čvor.

Rad bi takođe mogao da se unapredi i određivanjem brzine i efikasnosti konstrukcije stabala i pronalaženja k najbližih suseda prilikom primene drugih algoritama za konstrukciju samog k-d stabla ili loptastog stabla.

7. LITERATURA

[1] J. L. Bentley, “Multidimensional binary search trees used for associative searching,” Commun. ACM, vol. 18, no. 9, p. 509–517, Sep. 1975.

[2] R. A. Finkel and J. L. Bentley, “Quad trees a data structure for retrieval on composite keys,” Acta Informatica, vol. 4, no. 1, pp. 1–9, 1974.

[3] S. M. Omohundro, “Five balltree construction algorithms“, International Computer Science Institute Berkeley, 1989.

[4] A. Beygelzimer, S. Kakade, and J. Langford, “Cover trees for nearest neighbor,” Proceedings of the 23rd international conference on Machine learning, pp. 97–104, 2006.

[5] A. Guttman, “R-trees: A dynamic index structure for spatial searching,” Proceedings of the ACM SIGMOD international conference on Management of data, pp. 47–57, 1984.

[6] A.W. Moore, “An intoductory tutorial on kd-trees“, Technical Report No. 209, Computer Laboratory, University of Cambridge, 1991.

Kratka biografija:

Milena Kovačević rođena je u Beogradu 1998. god. Fakultet tahničkih nauka u Novom Sadu, studijski program Računarstvo i automatika, upisala je 2017. godine. Nakon završenih osnovnih studija , 2021. godine, upisala je master akademske studije iz oblasti elektrotehnike i računarstva.

kontakt: kmilena98@gmail.com