Geometrijski algoritmi @ MATF

Klasterovanje tačaka

Petar Zečević

Opis problema

Dat je skup tačaka u ravni. Potrebno je podeliti taj skup u dva disjunktna podskupa tako da se maksimizuje rastojanje njihovih konveksnih omotača. Vremenska složenost algoritma treba da bude $O(n^3)$.

Ulaz: *skup od n tačaka u ravni*

Izlaz: *dva skupa tačaka, njihovi konveksni omotači i tačke koje čine distancu*

Naivno rešenje problema

Za svaki par tačaka iz skupa, radimo sledeće:

1. Podelimo tačke na dva skupa. Sve tačke koje su "levo" od prave koju čini trenutni par tačaka idu u jedan skup, a sve koje su "desno" idu u drugi.

2. Gremovim algoritmom odredimo konveksne omotače skupova.

3. Iterativno računamo rastojanja temena prvog konveksnog omotača sa ivicama drugog (I temena drugog sa ivicama prvog). Izaberemo minimalno takvo rastojanje i zapamtimo između kojih tačaka je to rastojanje.

Kada smo prošli kroz sve parove tačaka, uzimamo onu podelu koja ima maksimalno rastojanje tačaka za rešenje. Vremenska složenost je $O(n^4)$. Da bismo obišli sve parove tačaka treba nam $O(n^2)$. Za svaki par tačaka imamo: podelu po skupovima ($O(n)$), računanje konveksnih omotača ($O(n\log n)$) i računanje svih kombinacija distanci ($O(n^2)$). Dakle: $O(n^2)*(O(n) + O(n\log n) + O(n^2))$ je $O(n^4)$.

Bolje rešenje problema

Na početku napravimo mapu u kojoj čuvamo sortirane nizove tačaka. Jedan par u mapi je jedna od tačaka iz skupa i sortirani niz u kom su tačke sortirane po uglovima koje zaklapaju sa tačkom iz ključa. Zatim kao i u naivnom algoritmu prolazimo kroz sve parove tačaka. Radimo sledeće:

1. Podelimo tačke na dva skupa.

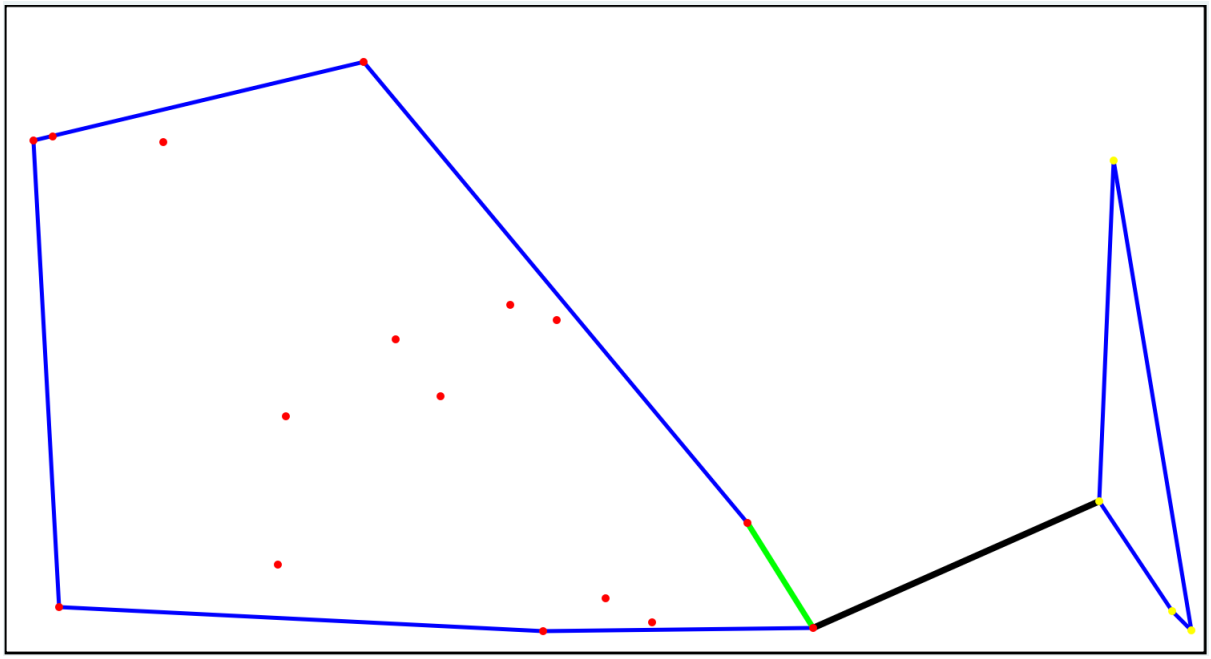
2. Gremovim algoritmom odredimo konveksne omotače skupova. Razlika je što sada ne sortiramo tačke već tražimo u mapi ono sortiranje koje nam odgovara.

3. Uzmemo prvu tačku prvog niza i iz nje povučemo tangente na drugi konveksni omotač i pamtimo u kojim tačkama se presecaju. Izdvojimo podniz iz drugog omotača tako što za početak uzmemo tačku preseka sa prvom tangentom a za kraj uzmemo tačku preseka sa drugom tangentom. Isto to uradimo i za prvim omotačem. Sada u petlji krećemo od prvih duži u podnizovima (odnosno parova prvih i drugih tačaka) i tražimo minimalno rastojanje od tih duži, narednih i prethodnih (ukoliko postoje). Kada je minimalno rastojanje u stvari rastojanje trenutnih duži, završavamo petlju i minimalno rastojanje proglašavamo za rastojanje omotača.

Kada smo prošli kroz sve parove tačaka, uzimamo maksimalno rastojanje tačaka za rešenje. Vremenska složenost je $O(n^3)$. Na početku pravimo mapu svih sortiranja po tačkama što je složenosti O(n2logn). Da bismo obišli sve parove tačaka treba nam $O(n^2)$. Za svaki par tačaka imamo: podelu po skupovima (O(n)), računanje konveksnih omotača s tim što već imamo sortirane tačke (O(n)), nalaženje tangenti (O(n)) i lokalna pretraga rastojanja (O(n)). Dakle: $O(n^2 \log n) + O(n^2) * (O(n) + O(n) + O(n) + O(n))$ je $O(n^3)$.

Vizuelizacija algoritma

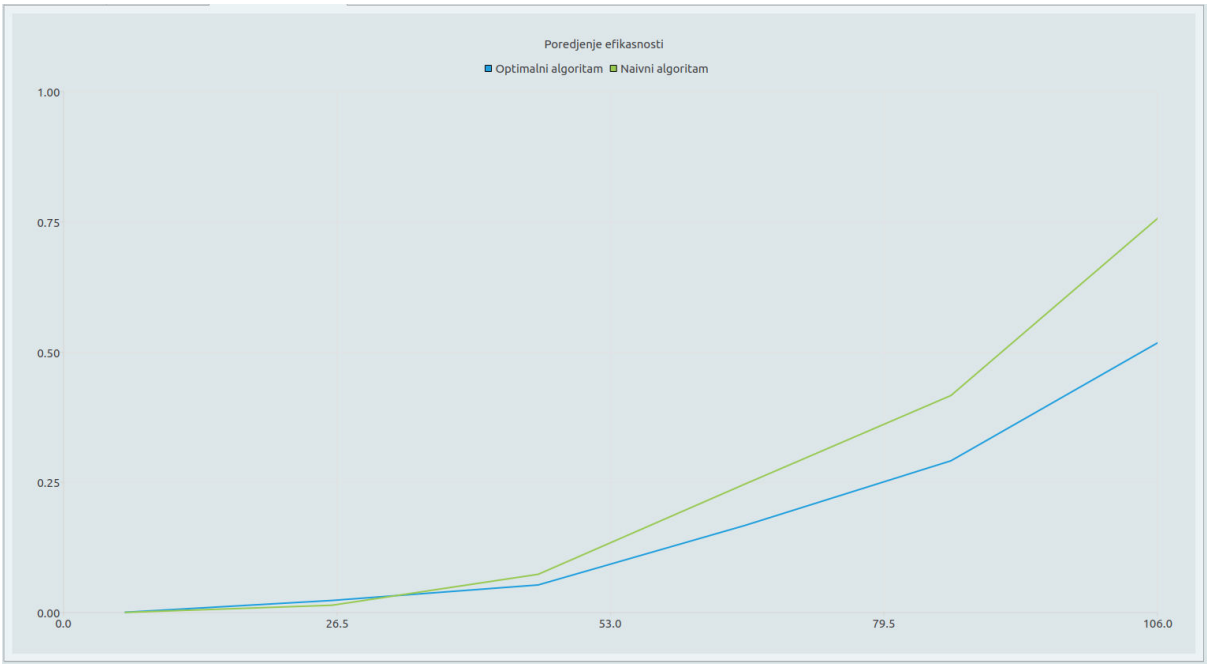
Jedan korak vizualizacije je prikaz jedne podele skupa na dva podskupa. Poslednji korak vizualizacije je prikaz one podele koja daje maksimalno rastojanje ta dva podskupa. Prikazane su tačke, konveksni omotači, duž po kojoj delimo skup u dva podskupa, i duž koja je u stvari distanca između omotača.



Poređenje efikasnosti naivnog i naprednog algoritma

Na grafu je prikazano poređenje naivnog i optimalnog algoritma. Možemo primetiti da je naivni algrotiram za dovoljno mali ulaz malo bolji od optimalnog. To je verovatno zato što se u optimalnom radi jedna vrsta pripreme a

i ima dosta poređenja za jednu iteraciju petlje. Za mali broj tačaka ta poređenja su verovatno suvišna, ali što se više povećava broj tačaka, te provere su isplativije.



Testiranje ispravnosti algoritma

Naziv testa	Opis testa	Ulaz	Očekivani izlaz
prviNasumičniTest	10 nasumično izabranih tačaka	Niz dimenzije 10	Poklapanje rezultata naivnog i naprednog algoritma
drugiNasumičniTest	20 nasumično izabranih tačaka	Niz dimenzije 20	Poklapanje rezultata naivnog i naprednog algoritma
treciNasumičniTest	3 0 nasumično izabranih tačaka	Niz dimenzije 30	Poklapanje rezultata naivnog i naprednog algoritma
datoteka1	Test sa tačkama koje čine pravu paralelnu y osi	Niz dimenzije 10	Poklapanje rezultata naivnog i naprednog algoritma

datoteka2	Običan test sa 8 tačaka	Niz dimenzije 8	Poklapanje rezultata naivnog i naprednog algoritma
datoteka3	Test sa tačkama koje se ponavljaju	Niz dimenzije 10	Poklapanje rezultata naivnog i naprednog algoritma
datoteka4	Običan test sa 9 tačaka	Niz dimenzije 9	Poklapanje rezultata naivnog i naprednog algoritma
nedovoljnoTacaka	Test u kom je dat nedovoljan broj tačaka	Niz dimenzije 3	Izlaz algoritma je 0
rastojanjeTackeOdDuzi	Test u kome su date tačka i duž i treba da se izračuna njihovo rastojanje	3 tačke	Poklapanje vrednosti rastojanja i najbliže tačke na duži sa tačnim vrednostima
rastojanjeTackeOdDuziParalelneYOsi	Test u kome su date tačka i duž i treba da se izračuna njihovo rastojanje, a duž je paralelna y osi	3 tačke	Poklapanje vrednosti rastojanja i najbliže tačke na duži sa tačnim vrednostima
rastojanjeTackeOdDuziNaKojuNemaNormalu	Test u kome su date tačka i duž i treba da se izračuna njihovo rastojanje, a tačka koja se dobije povlačenjem normale na pravu ne pripada duži	3 tačke	Poklapanje vrednosti rastojanja i najbliže tačke na duži sa tačnim vrednostima