SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO–MATEMATIČKI FAKULTET
MATEMATIČKI ODSJEK

Mihael Petrinjak

SEGMENTIRANA METODA NAJMANJIH KVADRATA

Zagreb, prosinac 2022.

Sadržaj

Sadržaj ii

Uvod 1

1 Problem 2
1.1 Motivacija 2
1.2 Matematička formulacija 3

2 Algoritam 4
2.1 Princip optimalnolsti 4
2.2 Pseudokod 6
2.3 Složenost algoritma 7

Bibliografija 8

Uvod

Ovaj rad je projektni zadatak za kolegij *Oblikovanje i analza algoritama*. Znanje preuzeto iz [1].

Poglavlje 1

Problem

1.1 Motivacija

Klasičan problem u analizi podataka je aproksimacija točaka u ravnini pravcem. Ima dobro nam poznato rješenje koje se efikasno računa iz danih podataka.
Neka je P skup točaka $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$ takav da $x_1 < x_2 < ... < x_n$. Znamo da funkciju

$$E(L, P) = \sum_i (y_i - ax_i - b)^2 \tag{1.1}$$

minimiziraju

$$a = \frac{n \sum_i x_i y_i - (\sum_i x_i)(\sum_i y_i)}{n \sum_i x_i^2 - (\sum_i x_i)^2} \qquad b = \frac{\sum_i y_i - a \sum_i x_i}{n} \quad . \tag{1.2}$$

Skup P ne mora biti lijep. Može se dogoditi da točke ne izgledaju kao da leže u blizini nekog pravca ili da postoji nekoliko pravaca koji dobro aproksimiraju zadani skup. Naravno, želimo formulaciju koja ne zahtjeva vizualizaciju podataka kako bismo odredili optimalan broj pravaca.

1.2 Matematička formulacija

Kao i gore $P = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\} = \{p_1, p_2, ..., p_n\}$ i $x_1 < x_2 < ... < x_n$. **Segment** u P je skup oblika $\{p_i, p_{i+1}, ..., p_j\}$ gdje $1 \leq i \leq j \leq n$. Tražimo particiju F skupa P koja se sastoji isključivo od segmenata u P. Za svaki $S \in F$ računamo minimalnu pogrešku aproksimacije po formulama (1.2) i (1.1).
Penalizacija particije F se definira kao suma sljedećih izraza.

(i) Broj segmenata u F pomnožen s konstantom $C > 0$.

(ii) Za svaki segment $S \in F$ vrijednost $E(L, S)$ gdje je L pravac dobiven *metodom najmanjih kvadrata*.

$E(L, S)$ je minimalna greška za proizvoljni pravac na segmentu S. Povečanjem broja segmenata raste vrijednost izraza (i) i pada vrijednost (ii). za smanjenje broja segmenata vrijedi obratno. Konstanta C modelira cijenu dodavanja segmenta u particiju. Veći C znači manji broj segmenata u rješenju.
Cilj *Segmentirane metode najmanjih kvadrata* je naći particiju minimalne penalizacije.

Poglavlje 2

Algoritam

2.1 Princip optimalnosti

Neka $OPT(n)$ predstavlja optimalno rješenje na točkama $p_1, ..., p_n$ i neka je $e_{i,j}$ vrijednost $E(L(a, b), \{p_i, ..., p_j\})$ gdje su a, b dobiveni iz (1.2), tj. minimalna greška aproksimacije pravcem. Primijetimo sljedeće.

Slutnja 2.1.1. *Ako je zadnji segment optimalne particije $\{p_i, ..., p_n\}$, onda je vrijednost optimalnog rješenja jednaka $OPT(n) = e_{i,n} + C + OPT(i - 1)$*

Problem je time sveden na manji. Primjenom istog pristupa opravdavamo sljedeću generaliziranu rekurziju.

Teorem 2.1.2. *Optimalno rješenje podproblema na $\{p_1, ..., p_j\}$ dano je s*

$$OPT(j) = \min_{1 \leq i \leq j}(e_{i,j} + C + OPT(i - 1)) \tag{2.1}$$

uz početni uvjet $OPT(0) = 0$.

Dokaz. Jakom indukcijom po j.

Baza:

$$OPT(1) = e_{1,1} + C + OPT(0) = 0 + C + 0 = C$$

Budući da je najmanji mogući broj segmenata jednak jedan, ovo je minimalna vrijednost za $\{p_1\}$.

Pretpostavka: Optimalno rješenje na $\{p_1, ..., p_k\}$ dano je s

$$OPT(k) = \min_{1 \leq i \leq k}(e_{i,k} + C + OPT(i - 1)),$$

za svaki $k \in \mathbb{N}, k < j$

Korak:

$$OPT(j) = \min \begin{cases} e_{1,j} + C + OPT(0), \\ e_{2,j} + C + OPT(1), \\ \quad\vdots \\ e_{j,j} + C + OPT(j-1) \end{cases} \tag{2.2}$$

Jedina odluka je postavljanje indeksa i na odgovarajuće mjesto. U svakom retku izraza (2.2) se po pretpostavci nalazi minimalna vrijednost penalizacije lijevih segmenata za takav izbor indeksa i. Izbor najmanje vrijednosti daje optimalno rješenje za $p_1, ...p_j$.

$\square$

2.2 Pseudokod

Algorithm 1 Segmentirana metoda najmanjih kvadrata

```
procedure SMNK(P, n, C)
    Polje OPT[0...n]
    OPT[0] ← 0
    Matricu E[1...n][1...n] inicijaliziraj na (0).
    for svi parovi i ≤ j do
        E[i][j] ← e_{i,j}
    end for
    for 1 ≤ j ≤ n do
        Računaj OPT[j] po rekurziji u teoremu 2.1.2
    end for
return OPT
end procedure

procedure IZVADI SEGMENTE(OPT, j, C)
    if j = 0 then
        Gotovo.
    else
        Pronađi i koji minimizira e_{i,j} + C + OPT[i − 1].
        return {p_i, ..., p_j}
        return IZVADI SEGMENTE(OPT, i − 1, C)
    end if
end procedure

procedure MAIN
    P, n, C    zadani.
    OPT ← SMNK(P, n, C)
    particija ← IZVADI SEGMENTE(OPT, n, C)
end procedure
```

2.3 Složenost algoritma

Vrijeme potrebno za računanje (1.2) i (1.1) je $\Theta(n)$. Dakle, za $e_{i,j}$ potrebno je $\Theta(n)$ operacija. Parova $1 \leq i \leq j \leq n$ je $\Theta(n^2)$. Nakon $O(n^3)$ vremena imamo sve vrijednosti $e_{i,j}$. Složenost traženja minimuma u devetom retku algoritma traje $O(n)$ i to za n vrijednosti parametra j. Složenost SMNK je $O(n^2)$.
Svako izvlačeje segmenata bez rekurzivnog poziva traje $O(n)$. U najgorem slučaju $\frac{n}{2}$ puta pa segmente pronalazimo u $O(n^2)$ vremena.

Bibliografija

[1] Jon Kleinberg i Eva Tardos, *Algorithm design*, Addison Wesley, 2006.