

Verjetnost sekanja konveksne podmnožice -
Poročilo

Peter Dolenc, Jan Rems

2017

1 Uvod

V nalogi se bova soočila z naslednjim problemom. Zamislamo si, da imamo na Evklidski ravnini neko *konveksno množico* C , ki vsebuje *konveksno podmnožico* C' . Najina naloga bo določiti verjetnost dogodka, da naključna premica, ki seka množico C seka tudi podmnožico C' . Gre za klasični problem s področja stohastične geometrije, ki se ga da posplošiti tudi na druge veje verjetnostne teorije. *Hipoteza*, ki jo postavlja trdi, da bo verjetnost zgoraj omenjenega dogodka enaka razmerju obsegov konveksnih množic. Nalogo sva razdelila, na dva dela. V prvem delu, problemu najprej postaviva *teoretični* okvir in rešitev izpeljeva analitično. Nato pa se problema lotiva še /eksperimentalno in spiševa program, ki se reševanja problema loti numerično.

2 Teoretična opredelitev problema in analitična izpeljava

Za začetek vpeljimo nekaj pojmov, ki so potrebni za formuliranje klasičnega izreka s področja integralske geometrije, ki nam bo v pomoč pri izpeljavi rezultata, ki ga predvideva najina hipoteza.

Definicija 1 *Množico vseh neorientiranih premic na \mathbb{R}^2 definiramo kot:*

$$\{(p, \theta) | 0 \leq \theta \leq 2\pi, p \geq 0\},$$

kjer p predstavlja oddaljenost premice od izhodišča, θ pa kot premice glede na x os.

S tem ko imamo definirano množico vseh premic na \mathbb{R}^2 , lahko na tej množici podamo *mero*.

Definicija 2 *Naj bo S neka množica premic v \mathbb{R}^2 , kjer so premice podane na način, kot je naveden v Definiciji 1. Potem na množici S definiramo mero $\mu : \mathbb{R}^2 \rightarrow [0, \infty]$, ki je podana s naslednjim predpisom:*

$$\mu(S) = \int \int_S d\theta dp$$

Naj dodamo, da je mera μ invariantna glede na toge premike.

Sedaj pa podajmo še naslednji pomembni izrek.

Izrek 1 (Cauchy-Croftonova formula) *Naj bo c regularna ravninska krivulja in $n_c(p, \theta)$ število točk, pri katerih premica, parametrizirana na način, kot je podan v Definiciji 1, seka krivuljo c . Potem za dolžino krivulje $L(c)$ veja:*

$$L(c) = \frac{1}{2} \int \int n_c(p, \theta) d\theta dp$$

Zaradi zgornjega rezultata je iskanje preseka premice s konveksno množico smiselno prevesti na problem iskanja presečišč premice in *sklenjene regularne krivulje*, ki omejuje dano konveksno množico. Tako nas namesto razmerja obsegov zanima *razmerje dolžine krivulj*, kjer ena omejuje konveksno množico C , druga pa njeno konveksno podmnožico C' . Tudi krivulji bomo ustrezno poimenovali γ in γ' . Krivuljam, ki omejujejo konveksne množice, pravimo *konveksne krivulje*.

Označimo z Γ dogodek, da naključno izbrana premica l seka krivuljo γ tj. $\Gamma = \{l \cap \gamma \neq \emptyset\}$, z Γ' pa dogodek da premica l seka vsebovano krivuljo γ' tj. $\Gamma' = \{l \cap \gamma' \neq \emptyset\}$. Verjetnost dogodka, da naključna premica, ki seka krivuljo γ , seka tudi γ torej izrazimo na naslednji način

$$P(\Gamma'|\Gamma) = \frac{P(\Gamma' \cap \Gamma)}{P(\Gamma)} = \frac{P(\Gamma')}{P(\Gamma)} \quad (1)$$

Ker je, če imamo opravka s sklenjeno konveksno krivuljo c , število presečišč premice s krivuljo c skoraj gotovo enako 0 oziroma 2, in pa zaradi Cauchy-Croftonove formule iz Izreka 1, velja naslednja enakost

$$\mu(\{L : L \cap c \neq \emptyset\}) = \int \int_{\{L : L \cap c \neq \emptyset\}} d\theta dp = L(c) \quad (2)$$

Če torej mero μ ustrezno normiramo, da postane verjetnostna z združitvijo enačb 1 in 2 hitro dobimo željeni rezultat

$$P(\Gamma'|\Gamma) = \frac{L(\gamma')}{L(\gamma)} \quad (3)$$

Sedaj ko smo intuitivno analitično dokazali postavljeno hipotezo, si oglejmo še eksperimentalni del naloge.

3 Eksperimentalno delo

3.1 Shema eksperimenta

Eksperimentalnega dela projekta sva se lotila z metodo *Monte Carlo*. Z generiranjem velikega števila naključnih premic in veliko ponovitvami poizkusa, sva želela potrditi postavljeno hipotezo, oz. se čim bolj približati rezultatu, ki jo le-ta napoveduje. Ker je naša mera μ invariantna za toge premike, je vseeno, kje v \mathbb{R}^2 se nahajata konveksni krivulji. Zato zadostuje, da delamo zgolj z krivuljami, ki so generirane simetrično glede na izhodišče koordinatnega sistema, oziroma so tako generirani vhodni podatki, ki množice določajo. Ideja je sledeča: *generirati dve naključni konveksni krivulji* γ in γ' , kjer je γ' strogo vsebovana v γ in pa *naključno premico*, ki seka γ ter ugotoviti, ali premica seka tudi γ' . Z mnogo ponovitvami implementiranega algoritma, bomo lahko *ocenili* verjetnost, da naključna premica, ki seka večjo konveksno krivuljo, seka tudi manjšo. Eksperiment sva razdelila na štiri podnaloge:

- generiranje premic

- generiranje konveksnih krivulj
- implementacija algoritma, ki analizira sekanje premic s krivuljami
- analiza dobljenih rezultatov in učinkovitosti algoritma

Simulacije geometrijskega problema sva se lotila postopoma. Na začetku sva poskušala preveriti najino hipotezo na enostavnejših konveksnih krivuljah - *elipsah*, kasneje pa sva obravnavala konveksne krivulje *splošnejše* oblike. Pri delu z elipsami je bil cilj napisati program, ki je sposoben obvladovanja osnovnih ukazov: *generiranja elips in premic* ter prepoznavanja, kdaj te premice res sekajo naše objekte. Ta orodja so nama pomagala pri nadaljnjem delu, ko sva, kot že omenjeno, z generiranjem naključnih konveksnih krivulj problem *posplošila*.

Drugi del izziva je predstavljal *generiranje naključnih premic*. Za to obstaja več načinov. Eden takšnih je tak, kot je opisano v definiciji: naključno določimo par (p, θ) , kjer p predstavlja oddaljenost premice do izhodišča, θ pa kot med premico in x osjo. Ker so naju zanimale le tiste premice, ki sekaj večjo od krivulj se pravi γ , sva poizkusila tudi z *alternativnim pristopom* generiranja naključnih premic in sicer z določitvijo parov *naključnih točk* znotraj večje konveksne krivulje ter s tem premice, ki skozi njih potekajo. Tako sva dobila generirane le premice, ki z gotovostjo sekajo γ in sva zato morala preveriti zgolj, ali sekajo tudi γ' . Zanimalo naju je tudi, kako se rezultati spremenijo ob drugačnem generiranju naključnih premic.

Po opravljenih simulacijah naju je, poleg *ujemanja* eksperimentalnega rezultata s postavljeno hipotezo, zanimalo tudi kako hitro dobljeni približek *konvergira* k točnemu rezultatu, ki sva ga dobila z analitičnim pristopom. Delo je potekalo v programskem okolju *Matlab*.

3.2 Generiranje premic

Pri projektu se je izkazalo, da je zelo pomembno, kako generiramo naključne premice. Da dobimo željeni rezultat, je potrebno generirati par (p, θ) iz enakomerno zvezne porazdelitve, kjer p predstavlja oddaljenost iz izhodišča in teče med 0 in najbolj oddaljeno točko večje konveksne krivulje od izhodišča, θ pa predstavlja kot med premico in x -osjo ter teče med 0 in 2π . Matriko velikosti $n \times 2$ z n pari takih parametrov sva potem s pomožno funkcijo spremenila v matriko velikosti $n \times 2$, kjer je prvi stolpec vseboval koeficiente k , drugi pa začetne vrednosti n iz eksplicitnega zapisa premic $y = k \cdot x + n$. To nama je omogočilo, da sva določila daljice s krajišči izven večje krivulje, s katerimi sva sekala obe krivulji.

Poskusila sva tudi alternativno: generirala sva pare točk v večji konveksni množici in skozi njih potegnili premice. Izkazalo se je, da tak način generiranja premic ne pripelje do istega rezultata, saj je že za elipse v središču prišlo do velike napake. Da so premice generirane enakomerno glede na parametre p in θ se je torej izkazalo za ključno.

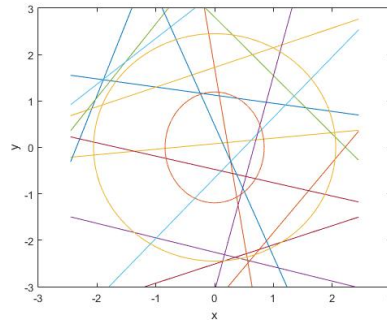
3.3 Generiranje konveksnih krivulj

Pri generiranju konveksnih krivulj sva se odločila da iz enostavnejših konveksnih krivulj preideva na zahtevnejše in splošnejše primere. Začela sva z generiranjem elips, kjer je tudi notranja elipsa kocentrična. Nato sva obnašanje algoritma

želela preveriti na primerih, kjer notranje krivulje niso nujno kocentrične, torej se nahajajo na poljubnem mestu znotraj zunanje krivulje. Zaradi olajšanja konstrukcije takih krivulj sva se odločila za delo s krožnicami. Nazadnje pa sva zgenerirala še naključni oziroma poljubni konveksni krivulji.

3.3.1 Elipse

Za generiranje elips sva napisala funkcijo “elipsa”, ki za vhodne podatke dobi parametra $xmax$ in $ymax$, ki določata največji možni števili a in b , ki se pojavita v parametriziranem zapisu elipse $x = a \cos(t), y = b \sin(t), t \in [0, 2\pi)$. Kot izhod vrne dva poligona točk v obliki elipse z naključno izbranimi parametroma $a \in (0, xmax)$ in $b \in (0, ymax)$ iz enakomerne zvezne porazdelitve za veliko elipso in $a_m \in (0, a)$ in $b_m \in (0, b)$ za malo elipso. V zadnjem koraku sva združili funkciji, ki generirata premice in elipsi ter dodala funkcijo, ki šteje presečišča med krivuljami. Za lažjo predstavbo si oglejmo sliko 1:



Slika 1: Primer generiranih elips in premic

Na sliki 1 vidimo 2 elipsi generirani s funkcijo “elipsa” s parametroma $xmax = 3$ in $ymax = 3$ in 15 premic z naključnima parametroma (p, θ) kot opisano zgoraj. V zadnjem koraku sva uporabilo funkcijo, ki šteje presečišča premic z obema elipsama ter število presečišč z manjšo krivuljo delila s številom presečišč z večjo. Rezultat sva primerjala z oceno razmerij obsega.

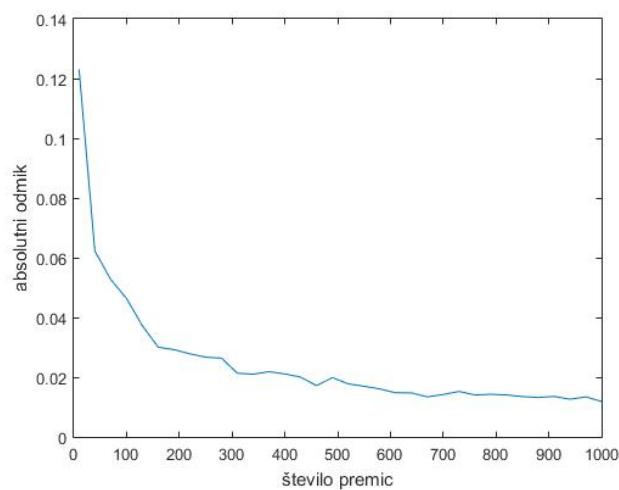
3.4 Analiza dobljenih rezultatov

Razmeroma zanesljive rezultate sva dobila, če sva preko posameznega para konveksnih krivulj generirala več kot 300 premic. Zato sva generirala 1000 parov elips in 400 parov konveksnih krivulj ter za vsak par 1000 premic in preštela presečišča z obema krivuljama. Rezultate sva shranila v matriko velikosti 1000×4 za elipse, ki po vrsticah prikazujejo posamezen poskus, po stolpcih pa razmerje obsegov, razmerje presečišč, absolutno razliko in predznačeno razliko oziroma matriko 400×5 za splošne krivulje, ki vsebuje dodaten stolpec: števila sekanja večje krivulje. Zanimalo naju je tudi, kako hitra je konvergenca najine metode, oz. koliko premic je potrebno generirati za posamezen par elips, da bo rezultat razmeroma točen. Rezultate sva dobila tako, da sva za vsak par konveksnih krivulj primerjala absolutno razliko do točnega rezultata po vsakih 30 dodatnih premicah.

3.4.1 Elipsa

Povprečje absolutnih vrednosti razlike med razmerjem obsegov in razmerjem premic, ki sekajo eno in drugo krivuljo je bila 0.0125, če pa dovolimo predznačeno razliko, se je v povprečju zmanjšala na 0.000046¹. Standardni odklon razlik je bil TUKI VSTA STANDARDNI ODKLON. Rezultati se skladajo z najinimi pričakovanji, morda je nekoliko presenetljiv velik standardni odklon, kar pomeni, da metoda ne deluje najboljše za posamezne krivulje, točne rezultate dobimo šele v povprečju.

Graf 2 prikazuje konvergenco metode na primeru elips.



Slika 2: Graf napake v razmerju s številom premic

¹Celotna tabela rezultatov je dosegljiva na github repozitoriju