

Quasi-Monte Carlo

Matej Marko

30. marca 2011

Úvod

Pri odhadovaní hodnoty integrálu pomocou metódy Monte Carlo vygenerujeme N bodov z definičného oboru (resp. z intervalu, na ktorom integrujeme) a odhad vypočítame ako priemer z funkčných hodnôt v týchto bodoch. Quasi-Monte Carlo sa zaoberá otázkou ako zvoliť tieto body.

Náhodný výber vzoriek je výhodný hlavne z pohľadu teórie (dokazovanie nestrannosti estimátora). V praxi má však tendenciu vytvárať zhluky vzoriek, resp. nepokryté oblasti. Ak sa teda stane, že funkcia má výrazný príspevok práve v nejakej nepokrytej oblasti, bude výsledný odhad nepresný.

Druhou možnosťou je použiť vzorky, ktoré sú deterministicky generované podľa určitého predpisu. Nakoľko neplatí predpoklad náhodnosti vzoriek, je potrebné upraviť teóriu okolo estimátora. V praxi však má tento prístup dobré výsledky, práve vďaka tomu, že rovnomernejšie pokrýva požadovaný interval.

Diskrepancia

Majme s -dimenzionálnu “tehlu”, ktorá je definovaná pomocou charakteristickej funkcie:

$$\chi(\mathbf{z}) = \begin{cases} 1 & \text{ak } 0 \leq \mathbf{z}|_1 \leq v_1, 0 \leq \mathbf{z}|_2 \leq v_2, \dots, 0 \leq \mathbf{z}|_s \leq v_s \\ 0 & \text{inak} \end{cases}$$

Objem takto definovanej tehly môžeme jednoducho spočítať pomocou hraničných hodnôt v_i :

$$V(A) = \prod_{i=1}^s v_i$$

Štandardným Monte Carlo estimátorom odhadneme objem tehly nasledovne:

$$\frac{m(A)}{N} = \frac{1}{N} \sum_{i=1}^N \chi(\mathbf{z}_i)$$

$m(A)$ - počet vzoriek, ktoré sa nachádzajú vo vnútri tehly
 N - počet všetkých vygenerovaných vzoriek

Naším cieľom je, aby odhad $\frac{m(A)}{N}$ bol čo najbližšie k skutočnému objemu tehly $V(A)$. To závisí od toho, ako rovnomerne sú vzorky rozmiestnené. Ak by vznikol zhluk vzoriek vo vnútri tehly, odhadnutý objem by bol príliš vysoký. Naopak pri zhluku mimo tehlu bude odhad nízky.

Diskrepancia bude veličina charakterizujúca kvalitu postupnosti vzoriek v zmysle ich rovnomerného rozmiestnenia:

$$D^*(\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N) = \sup_A \left| \frac{m(A)}{N} - V(A) \right|$$

Suprémum prebieha cez všetky možné tehly A . Je to teda maximálna možná chyba MC odhadu. Čím je hodnota diskrepancie menšia, tým je postupnosť vzoriek rozložená rovnomernejšie. Nutnou podmienkou pre postupnosť vzoriek je: $D^* \rightarrow 0$ pre $N \rightarrow \infty$, inak je táto postupnosť úplne nezmyselná pre akýkoľvek odhad.

Význam tejto veličiny ukazuje Koksma-Hlawka nerovnosť, ktorá obmedzuje chybu odhadu pomocou variácie funkcie a diskrepancie postupnosti vzoriek:

$$\left| \int f - \hat{I} \right| \leq \nu_{HK} \cdot D^*(\mathbf{z}_1, \dots, \mathbf{z}_n)$$

Koksma-Hlawka nerovnosť platí iba pre funkcie s konečnou variáciou. QMC prístup funguje aj pre funkcie s nekonečnou variáciou, ale uvedená nerovnosť sa nedá použiť pre obmedzenie chyby odhadu.

Variácia funkcie je veličina, ktorá hovorí, “ako veľmi sa funkcia mení”. V Koksma-Hlawka nerovnosti sa používa Hardy-Krauseova variácia:

$$\nu_{KH}(f(u, v)) = \int_0^1 \int_0^1 \left| \frac{\partial^2 f(u, v)}{\partial u \partial v} \right| du dv + \int_0^1 \left| \frac{\partial f(u, 1)}{\partial u} \right| du + \int_0^1 \left| \frac{\partial f(1, v)}{\partial v} \right| dv$$

Konstruktúra postupností

Jednoduchým ale pomalým algoritmom, ktorý generuje rovnomerne rozmiestnené vzorky je Lloyd relaxation. Algoritmus najprv vygeneruje náhodné vzorky, na ktorých opakuje nasledujúce kroky:

1. zostavíme Voronoi diagram
2. spočítame ťažiská buniek diagramu
3. presunieme vzorky do príslušných ťažísk

Výsledkom sú vzorky, ktoré sú rozmiestnené relatívne rovnomerne, ale stále náhodne. Nevýhodou je rýchlosť a tiež nutnosť poznať počet vzoriek dopredu.

Častým prístupom je snaha dosiahnuť Poissonove diskové rozdelenie, ktoré spĺňa vlastnosť, že žiadne dve vzorky nie sú bližšie ako daná vzdialenosť d . Naivný algoritmus generuje vzorky a pre každú testuje, či sa nachádza dostatočne ďaleko od už vygenerovaných vzoriek. Ak áno tak ju prijme, ak nie tak

ju zamietne. Pokračuje, až kým nevygeneruje požadovaný počet vzoriek. S pribúdajúcim počtom prijatých vzoriek, je však čoraz menej pravdepodobné, že náhodná vzorka bude prijatá. Algoritmus je teda opäť značne pomalý.

Algoritmus “najlepšieho kandidáta” aproximuje diskové rozdelenie, tak že generuje postupne sa zahusťujúcu Poissonovskú postupnosť vzoriek. Prvú vzorku vyberieme náhodne. Pre získanie vzorky $k + 1$ vygenerujeme $k \cdot q$ vzoriek, z ktorých vzberieme tú ktorá sa od doteraz prijatých nachádza najďalej. Parameter q určuje kvalitu vzoriek (čím viac bude kandidátov, tým väčšia bude pravdepodobnosť, že najlepší je dostatočne ďaleko).

Van der Corputova postupnosť

Van der Corputova postupnosť je zobrazenie, ktoré každému prirodzenému číslu priradí racionálne číslo z intervalu $[0, 1)$:

$$\begin{aligned} \Phi_b : \mathbb{N}_0 &\rightarrow \mathbb{Q} \cap [0, 1) \\ i = \sum_{j=0}^{\infty} a_j(i) b^j &\mapsto \Phi_b(i) = \sum_{j=0}^{\infty} a_j(i) b^{-j-1} \end{aligned}$$

Toto zobrazenie “vezme” zápis čísla i v sústave so základom b , otočí poradie cifier a pripíše na začiatok desatinnú čiarku. Na príklade so základom $b = 2$ vidíme, že vygenrované hodnoty $(0.5, 0.25, 0.75, 0.125, \dots)$ postupne uniformne zahusťujú interval $[0, 1)$.

Pri rozšírení do viacerých dimenzií je potrebné dávať pozor, aby súradnice neboli navzájom korelované.

Haltonova postupnosť

$$x_i = (\Phi_{b_1}(i), \dots, \Phi_{b_s}(i))$$

využíva pre každú súradnicu van der Corputovu postupnosť s prvočíselným základom. Použitie rôznych prvočísel zaručí nekorelovanosť súradníc.

Hammersleyho postupnosť

$$x_i = \left(\frac{i}{n}, \Phi_{b_1}(i), \dots, \Phi_{b_{s-1}}(i)\right)$$

je Haltonova postupnosť, so zmeneným predpisom pre prvú súradnicu. Túto postupnosť môžeme použiť v prípade, že dopredu vieme počet vzoriek, ktorý budeme generovať (n).

Poznámky na záver

Pri použití QMC je dôležité zabrániť vzniku korelácii, ktoré môže vzniknúť, ak sa rovnaká postupnosť použije na dvoch miestach. Tento fakt sa napríklad prejaví

pri použití rovnakej postupnosti pre každý pixel v path-tracery. Výsledok je z hľadiska číselnej chyby rovnako dobrý ako pri použití iných postupností, ale opticky vyzerá oveľa horšie, pretože pixely sú navzájom korelované.

Pri použití pravidelnej mriežky rastie očakávaná chyba exponenciálne s počtom dimenzií. Pri path-tracingu je počet dimenzií nekonečný, preto aj chyba bude nekonečná.