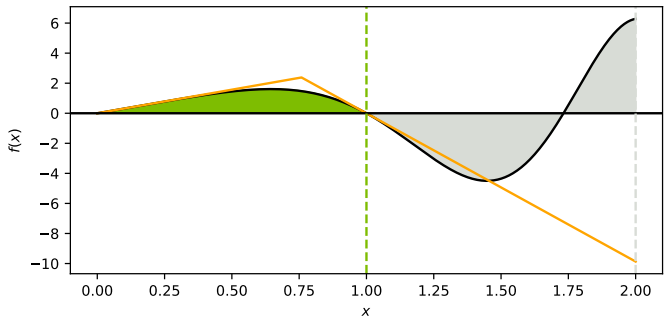
Statistiek en Gegevensverwerking met Python: Opdracht 3
Monte Carlo integratie

3 December 2019

Gegeven is de volgende functie

$$f(x) = \pi x \cos\left(\frac{\pi}{2}x^2\right)$$



1 Opdracht

1.1 Implementeer en vergelijk twee samplers

Genereer $N = 10000$ samples x_i, die volgens $f(x)$ met $x \in [0, 1]$ verdeeld zijn.

1. Implementeer eerst een 'hit and miss' algoritme. Kies hierbij zelf een $h(x)$ (zie definitie 10.3.2) die je geschikt lijkt (dit mag een uniforme zijn, of misschien iets zoals de oranje driehoek in de figuur).
2. Reken dan analytisch de cumulatieve $F(x) = \int_0^x f(x)\mathrm{d}x$ en haar inverse uit, en implementeer de methode van de inverse cumulatieve om samples te genereren.
3. Geef een korte beschrijving van je algoritmes en bewijs met een histogramplot van je gegenereerde samples dat de twee methoden werken. Neem ten slotte een kijkje naar de performantie van beide methoden
   - Wat is de efficientie ϵ van je 'hit and miss' methode? ($\frac{\text{aantal nuttige samples}}{\text{totaal aantal getrokken random getallen}}$. Voor de inverse cumulatieve is dit bij constructie gelijk aan 1.)
   - Wat is de snelheid van je methoden (aantal samples dat je kan generen per tijdseenheid)? **Opmerking:** Een simpele meting van N/tijd volstaat; er wordt niet gevraagd om te optimaliseren. Je mag dat natuurlijk proberen indien je dat leuk vindt, maar let daarbij op dat python geen snelle taal is.

1.2 Monte Carlo integratie

Maak een Monte Carlo schatting (met fout) van $F(u) = \int_0^u f(x)\mathrm{d}x$, voor $u = 1$, en voor $u = 2$.

1. Neem een kijkje naar het lijstje technieken in voorbeeld 10.5.1. Implementeer de ruwe Monte Carlo methode, en nog een andere techniek naar keuze (met eventuele aanpassingen indien nodig). Pas deze toe met $N = 100$ samples, en vergelijk de uitkomsten met de analytische waarde.

2. Plot de fouten op de uitkomsten van beide technieken (en voor beide integralen) als functie van het aantal samples N. Ga hiervoor van $N = 4$ tot ≈ 40000 in een aantal logaritmische stappen (vergroot N telkens met dezelfde factor). Je kan de fouten voor een bepaalde N schatten door de integratie pakweg 100 keer te herhalen met hetzelfde aantal samples (en verschillende random getallen natuurlijk).

1.3 2D toevalsgetallen

We gaan nu N gebeurtenissen te genereren binnen een cirkel met straal R, volgens de dichtheidsfunctie:

$$\rho(r; R)\mathrm{d}r = \frac{\pi r}{R^2} \cos\left(\frac{\pi r^2}{2R^2}\right) \mathrm{d}r$$

1. Trek een set poolcoördinaten (r, θ), en bereken dan de overeenkomstige (x, y)-coordinaten (r trek je uit ρ, θ mag uniform verdeeld zijn). Kies R verschillend van 1, zodat je kan tonen dat het algemeen werkt.

2. Maak met `numpy.histogram2d` een 2D histogram van de (x, y) data.

3. Neem dan een 1D doorsnede doorheen dit histogram (een lijn bins die doorheen de oorsprong passeert), en plot de binwaarden als functie van de straal. Je kan eventueel ook over meerdere doorsneden uitmiddelen. De nodige N om een mooie plot te krijgen zal afhangen van het aantal bins dat je kiest.

4. Zet dan de gegeven theoretische curve op dezelfde figuur. Pas op! Er is mogelijks nog een normalisatiefactor nodig om deze curve en het histogram overeen te laten stemmen.

2 Verslag

De deadline ligt op **15 December 23:59u**.

- Dien jullie verslag en code in als .pdf en .py bestanden.