Symulacje Komputerowe
Sprawozdanie 1

1 Ogólne uwagi

Sprawozdania można wykonywać w grupach dwu- lub trzyosobowych. Prace indywidualne są akceptowane, ale ze względu na ilość pracy doradza się połączenie sił. Ocena za kody, estetykę, poprawność, wnioski i prezentację jest przyznawana z równą wagą za każde zadanie oddzielnie. Jedynie terminowość liczy się oddzielnie.

Rozwiązanie każdego zadania powinno się składać ze wstępu teoretycznego, sformułowania problemu badawczego, propozycji rozwiązania problemu (tutaj znaleźć się powinny np. ważniejsze pseudokody), przedstawieniu wyników (np. odpowiednio dobrane wykresy i tabele) i wniosków, w których można podsumować wyniki, postawić ewentualne hipotezy badawcze, postawić pytania odnośnie ewentualnych kierunków dalszych prac. Do sprawozdania załączamy listę kodów. Kody do każdego zadania powinny się znaleźć w osobnym pliku .ipynb lub .m. W przypadku korzystania z Markdowna możliwe jest umieszczenie kodów w pliku ze sprawozdaniem, koniecznie pod rozwiązaniem danego zadania. Kody w kolejnych komórkach w Jupyterze powinny być modularne, tzn. deklaracje osobnych funkcji powinny być umieszczone w osobnych komórkach, podobnie wywołania dla różnych analiz też powinny się znaleźć w osobnych komórkach. Cała praca (kody+sprawozdanie) powinna zostać wysłana w formacie .rar i podpisana nazwiskami wszystkich członków grupy w następujący sposób: KaczyńskiTusk1.rar.

Do generowania liczb pseudolosowych z rozkładu jednostajnego wykorzystujemy własny kod z pierwszego zadania niniejszego sprawozdania. W przypadku pominięcia tego zadania, można wykorzystać generator napisany na laboratorium. Do generowania liczb z rozkładu wykładniczego wykorzystujemy kod z laboratorium.

W sprawozdaniu rozwiązania zadań powinny być samodzielne: nie korzystamy z kodów innych grup, można jednak inspirować się cudzymi ideami, czy wspólnie opracowywać pseudokody, dowody twierdzeń itp. (w takim wypadku należy napisać, do kogo należał pomysł).

Wskazówki co do pisania sprawozdań:

- Zbiór esejów nt. wizualizacji danych http://www.biecek.pl/Eseje/
- Ogólne uwagi ze strony (+linki) dra Płociniczaka: `http://prac.im.pwr.edu.pl/~plociniczak/doku.php?id=dyplomanci`
- Wskazówki ze strony pana Ślęzaka: `http://prac.im.pwr.wroc.pl/~slezak/rapWytyczne.pdf`
- *Czysty kod*, R. C. Martin

W sprawie pytań lub wątpliwości można pisać maile, łapać w Instytucie, w kinie, na Facebooku, w górach, gdzie bądź. Postaram się służyć pomocą. Warto konsultować postępy prac na bieżąco.

2 Generator liczb pseudolosowych

Zaimplementuj (30%) generator liczb pseudolosowych naturalnych z zakresu $1,...,M$ postaci

$$x_{k+1} = \sum_{i=1}^{k} a_i x_i^{p_i} \mod m, \tag{1}$$

gdzie $n > k$, $x_1,...x_k$ są ziarnem podawanym przez użytkowika, m i M są liczbami całkowitymi podawanymi przez użytkownika, a $(a_1,...a_k)$ oraz $(p_1,...p_k)$ wektorami podawanymi przez użytkownika. Jest to algorytm rekurencyjny:

$$x_{k+n} = \sum_{i=1}^{k} a_i x_{i+n}^{p_i} \mod m$$

Uwaga! Pamiętamy, że chcemy dostać liczby całkowite!

Następnie (40%) napisz konkretny generator liczb pseudolosowych (użytkownik podaje tylko M - zakres oraz N - ile liczb chce wygenerować), w którym ziarno będzie sczytywane z cputime oraz czujników temperatury komputera, a wektory a i p będą odpowiednio wybrane, przy czym $k > 2$, co najmniej dwie współrzędne wektora a są niezerowe, co najmniej jedno $p_i > 1$. Dla $M = 10000$ zbadaj częstość występowania każdej z liczb $0,1,2,...,M-1$. Zbadaj, czy jakieś liczby występują istotnie częściej niż inne.

Na podstawie powyższego generatora napisz genetator liczb z rozkładu $U(a,b)$, $a < b$. Zadbaj o to, aby rozdzielczość generatora była blisko epsilona maszynowego, tzn. nie istnieje liczba typu `double` z zadanego przedziału, której prawdopodobieństwo wylosowania byłoby równe 0. Przeprowadź testy statystyczne zgodności liczb wylosowanych przez Twój generator z rozkładem jednostajnym. Dobrze widziane będą testy inne niż poznane na zajęciach, jednak każdy z nich musi być krótko omówiony.

O generatorach liczb pseudolosowych można poczytać tutaj: `http://dlibra.itl.waw.pl/dlibra-webapp/Content/588/TiTI-2001_2_30.pdf`

3 Bryły Steinmetza

Uogólnij poznaną metodę Monte-Carlo szacowania całek na funkcje d-zmiennych, tzn. napisz funkcję, która oszacowuje całkę

$$\int ... \int_D f(x_1,x_2,...,x_d)dx_1dx_2...dx_d, \tag{2}$$

gdzie $f : \mathbb{R}^d \mapsto \mathbb{R}$, a $D \subset \mathbb{R}$ jest zbiorem ograniczonym. Jak szybko całka po $[0,1]^d$, $d = 1,2,3,4,5$ z funkcji $f_d(x_1,...,x_d) = x_1 + ... + x_d$ zbiega do prawidłowego wyniku? Wykonaj odpowiednie wykresy (podobnie jak w Zadaniu 4), postaw hipotezę o tempie zbieżności. (20% punktów za ten podpunkt)

Definicja 1. *2-Bryłą Steinmetza nazywamy bryłę, która powstaje jako przekrój dwóch walców o tym samym promieniu i wspólnym środku ciężkości, które leżą do siebie prostopadle. Analogicznie, 3-bryła Steinmetza to przecięcie trzech takich walców.*

Analityczne wyprowadzenie wzorów na objętość można znaleźć np. u Fichtenholza, Tom II, 343.11.

Spośród poniższych podpunktów wykonaj jeden wybrany (są różnie punktowane):

1. (80% punktów) Znając analityczny wzór na pole powierzchni kuli, 2- i 3- Bryły Steinmetza (albo wyprowadzić, albo podać referencje do artykułu, w którym znajduje się wyprowadzenie; w Wikipedii można znaleźć wzór dla dwucylindra), oszacuj metodą Monte-Carlo stosunek pola powierzchni do objętości podanych brył w zależności od promienia r kuli lub walca. Uwaga! Nad "wzorem przekazywanym do całki"trzeba pomyśleć! Jeśli używamy jakiegoś wzoru, należy opisać (udowodnić) swoją koncepcję lub podać referencje!

2. (55% punktów) Oszacuj objętość elipsoidy o półosiach a, b, c.

3. (30% punktów) Oszacuj objętość d wymiarowej kuli o promieniu r.

4 Generowanie zmiennych losowych z rozkładów α-stabilnych

Definicja 2. *Zmienne losowe α-stabilne definiujemy jako zmienne losowe, których funkcja charakterystyczna jest postaci*

$$\phi(\xi) = e^{i\mu\xi - |\sigma\xi|^{\alpha}(1 - i\beta\,\mathrm{sgn}(\xi)\tan\frac{\pi\alpha}{2})},\ \alpha \neq 1, \tag{3}$$

oraz

$$\phi(\xi) = e^{i\mu\xi - |\sigma\xi|^{\alpha}(1 - i\beta\,\mathrm{sgn}(\xi)\frac{\pi\log|\xi|}{2})},\ \alpha = 1. \tag{4}$$

$\alpha \in (0,2]$ *jest tu parametrem stabilności,* $\beta \in [-1,1]$ *parametrem skośności,* $\sigma > 0$ *parametrem skali, a* $\mu \in \mathbb{R}$ *parametrem lokalizacji.*

Rozkład stabilny będziemy oznaczali $S(\alpha,\beta,\mu,\sigma)$. Gdy $\alpha = 2, \beta = 0$, to otrzymujemy rozkład normalny, natomiast dla $\alpha = 1, \beta = 0$ - rozkład Cauchy'ego.

Generuj U z rozkładu U(-$\pi/2$,$\pi/2$) ;
Generuj niezależnie E z rozkładu wykładniczego $E(1)$;
if $\alpha \neq 1$ **then**
  $S = (1 + \beta^2 \tan^2 \frac{\pi\alpha}{2})^{1/2\alpha}$;
  $B = \frac{1}{\alpha}\arctan(\beta \tan\frac{\pi\alpha}{2})$;
  $X = S\frac{\sin(\alpha(U+B))}{(\cos(U))^{1/\alpha}}\left(\frac{\cos(U-\alpha(U+B))}{E}\right)^{\frac{1-\alpha}{\alpha}}$;
  return $Y = \sigma X + \mu$;
end
else
  $X = \frac{2}{\pi}\left((\frac{\pi}{2} + \beta U)\tan U - \beta\log\left(\frac{\frac{\pi}{2}E\cos U}{\frac{\pi}{2} + \beta U}\right)\right)$;
  return $Y = \sigma X + \frac{2}{\pi}\beta\sigma\log\sigma + \mu$;
end

Powyższy pseudokod został zaproponowany przez prof. R. Werona, jest szeroko cytowany w wielu miejscach.

Zadanie polega na napisaniu generatora liczb pseudolosowych z rozkładu stabilnego o zadanych parametrach. Dla $\alpha \in \{1/2, 1, 3/2, 2\}$ stwórz histogramy, dystrybuanty empiryczne. Dla $\alpha \in \{1, 2\}$ przetestuj zgodność wygenerowanych próbek (KS-test) z odpowiednimi rozkładami.

Uwaga: w części teoretycznej warto tutaj zagłębić się w teorię rozkładów stabilnych, nieskończenie podzielnych itd. Można używać innego algorytmu. W przypadku każdego algorytmu podać referencje.

5 Optymalizacja metody akceptacji-odrzucenia

Są trzy warianty zadania. Należy wybrać jeden.

Metodę akceptacji-odrzucenia do generowania zmiennych losowych z rozkładów o nośnikach zwartych można zoptymalizować w prosty sposób: za gęstość bazową wystarczy przyjąć gęstość zmiennej losowej, która zachowuje się „podobnie" jak gęstość zmiennej losowej, którą chcemy generować.

Lemat 1. *Jeśli U jest zmienną losową o rozkładzie $U(0,1)$, to zmienna losowa $V = U^n$ ma rozkład $B(1/n, 1)$.*

Udowodnij powyższy Lemat (15% punktów). Następnie przypomnij sobie metodę kompozycji z wykładu: jeśli mamy zmienną losową X o dystrybuancie $F_X(x) = \sum_{i=1}^{n} p_i F_{Y_i}(x)$, $p_i > 0$, $\sum_i p_i = 1$, gdzie F_{Y_i} są dystrybuantami niezależnych zmiennych losowych Y_i, to gęstość $f_X(x) = \sum_{i=1}^{n} p_i f_{Y_i}(x)$.

Wariant 1 (100%) Będziemy chcieli poprawić wydajność generowania metodą akceptacji-odrzucenia zmiennych losowych z rozkładu $f(x) = \frac{3}{2}\sin x \cos^2(x)$ na przedziale $[0, \pi]$ (do 85% punktów). Problem w dotychczasowej metodzie polegał na tym, że w kodzie korzystaliśmy z pętli `while`, w której warunek wyjścia z pętli statystycznie często nie jest spełniony. Jeśli zastanowić się nad przyczyną tego stanu rzeczy, dochodzimy do wniosku, że prawdopodobieństwo wyjścia z pętli można zwiększyć, jeśli gęstością bazową jest nie gęstość rozkładu jednostajnego, lecz gęstość g, która jest w jakiś sposób podobna do f. Używając powyższego Lematu, rozwinięcia funkcji f w szereg (np. Taylora, ale inne rozwinięcia (np. ortogonalne) mogą być lepsze!) lub metody kompozycji, zaproponuj szybszą metodę generowania zmiennych losowych z rozkładu o gęstości f zmiennymi losowymi z rozkładów jednostajnych. Uwaga na założenia!

Wskazówka: f można przybliżać albo wielomianami, albo funkcjami schodkowymi. W drugim wypadku skorzystanie z metody kompozycji wydaje się oczywiste: podzielmy odcinek $[0, \pi]$ na np. N równych części i przybliżajmy f gęstościami rozkładów jednostajnych o odpowiednich wagach.

Inny pomysł na optymalizację: w wielu językach programowania wykonanie pętli bywa bardziej czasochłonne niż operacje globalnie działające na jakichś strukturach danych. W języku `R` na przykład wykonanie funkcji `lapply` przeważnie bywa szybsze niż wykonanie jakiejś operacji na każdym elemencie listy w pętli. Wiemy, że na n wykonań pętli średnio otrzymamy n/c akceptacji. Pomysł dla Pythona jest następujący: w pierwszym kroku losujemy $c \cdot n$ liczb mniej więcej w taki sposób

$$\texttt{X} = \texttt{rand}(\texttt{c} * \texttt{n})$$

$$\texttt{Y} = [\texttt{rand}(\texttt{c} * \texttt{n}) <= \texttt{f}(\texttt{X})/(\texttt{c} * \texttt{h}(\texttt{X}))],$$

dostając średnio n liczb. A co, jeśli dostaniemy mniej lub więcej liczb?

Porównaj czas generowania zmiennych losowych zaproponowaną przez siebie metodą z metodą z laboratorium. Przeprowadź analizę zgodności rozkładów (histogram, test χ^2, wykres kwantylowy).

Grupy, które rozwiążą to zadanie poprawnie (tzn. bez błędów merytorycznych i w kodzie), będą oceniane względem skuteczności optymalizacji, mianowicie: grupa, która wygeneruje 100000 zmiennych losowych w najkrótszym czasie na komputerze prowadzącego lub uczelnianym (komputer przy wejściu do sali lub komputer w pokoju 6.06 C-11; jak widać, można mieć nawet 3 próby), zdobywa 85%, grupa z drugim wynikiem 80%, natomiast pozostałe grupy po 75%.

Wariant 2 (za 55% punktów) Zaimplementuj generatory liczb pseudolosowych z rozkładów: Cauchy'ego, χ-kwadrat, t-Studenta, Bernoulliego. Uogólnić Zadanie 8. z listy zadań dla dowolnego wektora $(p_1, \ldots p_n)$.

Wariant 3 (za 70% punktów) Korzystając np. z https://jakevdp.github.io/blog/2012/08/18/matplotlib-animation-tutorial/ , napisać kod, który pokaże na animacji zbieżność empirycznego p-tego momentu $\frac{X_1^p + X_2^p + \ldots + X_n^p}{n}$ do teoretycznego p-tego momentu $\mathbb{E}X^p$ dla wygenerowanej próby z rozkładu normalnego $N(0, \sigma)$. Napisać kod, który pokaże na animacji zbieżność dystrybuanty empirycznej zadanego rozkładu (normalnego, wykładniczego, trójkątnego) do dystrybuanty teoretycznej.

6 Uogólniony Problem Sekretarki

W klasycznym problemie sekretarek zakładamy, że przesłuchujemy kolejno N kandydatek o różnym poziomie umiejętności. Po przesłuchaniu każdej podejmujemy decyzję: przyjąć ją lub odrzucić. Jeżeli przyjmiemy ją, pozostałe odprawiamy. Jeżeli odrzucimy ją, nie możemy jej złożyć ponownie oferty. Celem gry jest znalezienie strategii gwarantującej wybór najlepszej kandydatki z najwyższym prawdopodobieństwem. Problem ten ma rozwiązanie ścisłe: należy przesłuchać $r = N/e$ pierwszych kandydatek, sprawdzić, jaki był najwyższy poziom umiejętności wśród nich, po czym spośród pozostałych $N - r$ kandydatek wybrać pierwszą, której poziom umiejętności będzie wyższy od wcześniej zbadanego maksimum. Prawdopodobieństwo znalezienia optymalnej kandydatki dla dużych N jest bliskie $1/e$.

Problem jest następujący. Przesłuchujemy kolejno 80 analityków, 100 programistek i 120 testerów o umiejętnościach będących liczbami losowymi z rozkładu $N(0,1)$. Odrzuceni kandydaci nie wracają. Potrzebujemy zatrudnić 3 analityków, 3 programistki i 3 testerów. Za pomocą metod Monte-Carlo opracuj strategię, która z dużym prawdopodobieństwem będzie nam zwracać najlepszych kandydatów, mianowicie

- zmaksymalizujemy prawdopodobieństwo, że wszyscy trzej analitycy będą najlepsi spośród grupy kandydatów

- zmaksymalizujemy średnią sumę umiejętności trzech zatrudnionych programistek
- zmaksymalizujemy prawdopodobieństwo, że wszyscy trzej zatrudnieni testerzy znajdą się wśród 10% najlepszych spośród kandydatów.

We wstępie teoretycznym do tego zadania można zamieścić dowód rozwiązania klasycznego problemu sekretarek. Jeśli znaleźliście jakieś ciekawe rozwiązanie analityczne, zacytujcie je.

Grupy, które rozwiążą to zadanie poprawnie (tzn. bez błędów merytorycznych i w kodzie), będą oceniane względem wyników, które otrzymają, przykładowo: grupa *A* wymyśliła strategię, która zatrudnia trzech najlepszych analityków z prawdopodobieństwem $1/20$, natomiast wszystkie grupy mają mniejsze prawdopodobieństwa. Najsłabiej poradziła sobie grupa *B*. Grupa *A* dostaje 100% punktów za ten podpunkt, grupa *B* 75%, a pozostałe grupy znajdą się oczywiście pomiędzy.