Grafy i sieci:
Generator sieci bezskalowej II (model Barabasiego-Albert)

Eryk Warchulski
Kanstantsin Padmostka
Prowadzący: dr inż. Sebastian Kozłowski

23 marca 2019
wer. 1.0

Spis treści

Streszczenie

Dokument ten zawiera szczegółowy opis zadania projektowego, który ma potwierdzić zrozumienie tematu przez autorów i składa się z trzech częsci.
Część pierwsza (1) składa się z dwóch sekcji. W sekcjach (1) i (2) znajduje się kolejno: omówienie tematu zadania oraz wskazanie algorytmu, który będzie implementowany. Część ta odpowiada sprawozdaniu numer 1 umieszczonemu w harmonogramie projektu.
Część druga (2) składa się z trzech sekcji. W sekcjach (3) i (4) znajduje się krótki opis teorii grafów, które są tematem tego projektu. Omawiane są kolejno grafy losowe w modelu Erdosa-Renyi oraz – najbardziej istotne z punktu widzenia projektu – grafy losowe w modelu Barabasi-Albert. Sekcja (5) zawiera informacje dotyczące implementacji algorytmu wskazanego w (2). Omówiona jest tam technologia użyta do implementacji szybkiego i przenośnego generatora sieci bezskalowych oraz planowane struktury danych. W sekcji (6) opisane są planowane eksperymenty numeryczne, które będą przeprowadzane w ramach realizacji projektu.
Część trzecia (3) będzie poświęcona omówieniom wyników eksperymentów numerycznych, jakości implementacji generatora sieci oraz sformułowane zostanę wnioski końcowe[1].

[1] Ta część dokumentu zostanie uzupełniona w przyszłości, tj. po implementacji oraz przeprowadzaniu eksperymentów.

Sprawozdanie 1
Opis zadania projektowego

1 Opis zadania

Postawionym przed nami zadaniem jest zaimplementowanie generatora grafów losowych w ujęciu Barabasiego-Albert (BA), które zostanie szczegółowo opisane w dalszej części dokumentu (3). Generator poza spełnianiem swojej podstawowej funkcji musi charakteryzować się przenośnością oraz jak najmniejszą złożonością obliczeniową i pamięciową. Dla poprawnie działającego generatora kolejnym krokiem w realizacji zadania jest zbadanie rozkładów stopni wierzchołków grafów i porównanie ich z modelami teoretycznymi. Pełna realizacja zadania zakłada istnienie możliwości zapisu wygenerowanego grafu do ustalonego formatu, co pozwoli odwtorzyć sam graf oraz przebieg eksperymentów numerycznych.

2 Implementowany algorytm

W celu zrealizowania projektu zostanie zaimplementowany algorytm generowania grafów w modelu BA z selekcją wierzchołków opartą o zmodyfikowaną regułę *koło ruletki* zwaną ROLL-tree [4] . Algorytm ten jest opisany w literaturze przedmiotu jako `BA-model with simpleRW` [5]. Na rysunku (1). znajduje się pseudokod tego algorytmu:

Data: n, n_0, m, $V = \emptyset$, $E = \emptyset$
Result: $\mathcal{G}(V, E)$
dodaj n_0 wierzchołków do grafu $\mathcal{G}$
for *($i = n_0 + 1$; n; $i = i + 1$)* **do**
  stwórz nowy wierzchołek v_i
  for *($k = 1$; m; $k = k + 1$)* **do**
    wylosuj bez zwracania wierzchołek na podstawie v_l ze zbioru $\{v_1, \ldots, v_{i-1}\}$
    na podstawie reguły *ROLL-tree* wykonaj:
    $E = E \cup \{(v_l, v_i)\}$
  end
end

Algorithm 1: Algorytm generowania sieci BA

Pseudokod algorytmu jest opisowy i poza wyjaśnieniem reguły tworzenia krawędzi między wierzchołkami nie wymaga komentarza. Reguła ta jest wyjaśniona w sekcji dotyczącej grafów losowych w modelu BA (4).
Złożoność obliczeniowa takiej metody wynosi $O(|E|(|V| + 1))$.

2.1 ROLL-tree

Naiwna reguła losowania - reguła koła ruletki okazuje się być nieoptymalna dla grafa o dużej ilości wierzchołków. W ramach projektu zostanie zaimplementowana metoda losowania **ROLL-tree** [4], oparta na "kubełkowaniu" wierzchołków o takim samym stopniu wierzchołków - że wierzchołki o takim samym stopniu mają takie same prawdopodobieństwo wylosowania, czyli możemy ich przechowywać w takiej samej strukturze - kubełku, co jest w istocie podzbiorem wierzchołków. Dla optymalizacji przeszukiwania, kubełki te są umeszczane w wyważonym drzewie

binarnym, co z kolej zezwala na losowanie z użyciem reguły "dziel i rządź"[2]. Pełny algorytm takiego losowania w postaci pseudokodu został umieszczony na rysunku 2

Data: Drzewo ROLL zawierające kubełki ROLL
Result: losowy wierzchołek v_k
$Rnode \leftarrow Tree.root$
while *Rnode nie jest kubełkiem* **do**
  Wylosuj jednostajnie $r \in [1...(Rnode.w_R + Rnode.w_L)]$
  if $r \leqslant Rnode.w_L$ **then**
    $Rnode \leftarrow Rnode.Lchild$
  else
    $Rnode \leftarrow Rnode.Rchild$
  end
end
$B_i \leftarrow bucket(Rnode)$
$v_k \leftarrow$ wybierz z prawdopodobieństwem jednostajnym element z B_i
return v_k

Algorithm 2: Algorytm losowania z drzewem ROLL

Projekt zostanie wykonany w języku `C++` oraz `R`. Więcej informacji znajduje się w drugiej cześci dokumentu (5).

[2] ang. divide and conquer

Sprawozdanie 2
Implementacja oraz eksperymenty numeryczne

3 Grafy losowe

Stosowanie teorii grafów do modelowania zjawisk zachodzących w realnym świecie jest oparte w dużej mierze na grafach losowych. Podyktowane jest to faktem, że zjawiska te i towarzyszące im zdarzenia wykazują w skali makroskopowej charakter stochastyczny. Przykłady dziedzin, w których stosowane są grafy losowe do modelowania pewnych zjawisk są następujące:

- sieci połączeń handlowych
- sieci `WWW`
- sieci neuronowe (rekurencyjne)
- sieci społecznościowe (np. `Facebook`)

Zdefiniowanie grafu losowego wymaga z kolei zdefiniowania struktur jak *przestrzeń grafów losowych* $\mathcal{G}$, która jest wyposażona w unormowaną miarę $\mathbb{P}(\bullet)$ mówiącą o prawdopodobieństwie wylosowania grafu G o pewnych właściwościach [3].
Zadanie to ze względu na złożoną strukturę obiektów jakimi są grafy nie jest tak intuicyjne jak określenie przestrzeni probabilistycznej dla zdarzeń, które można reprezentować liczbami. Z tego względu istnieje szereg alternatywnych modeli, które podejmują się rozwiązania tego zadania. Pokrótce zostanie omówiony najstarszy i najprostszy model wprowadzony przez Erdős'a i Rényi'e jeszcze w latach 60. ubiegłego wieku [2].

3.1 Model E-R

Model ten jest oparty o dwójkę parametrów (n, p): parametr $n \in \mathbb{N}$ oznacza liczbę wierzchołków generowanego grafu G, a $p \in (0, 1)$ stanowi o prawdopodobieństwie zdarzenia polegającego na zaistnieniu krawędź między każdą parą z n^2 wierzchołków grafu G.
Na podstawie powyższego łatwo widać, że rozkład stopni wierzchołków w grafie zadany jest przez rozkład dwumianowy z funkcją gęstości prawdopodobieństwa

$$p(n, k; p) = \binom{n-1}{k} p^k (1-p)^{n-k-1} \tag{1}$$

implikuje to fakt, że średni stopień wierzchołka $\mathbb{E}deg(v)$ wynosi $(n-1)p$. Ponadto, prawdopodobieństwo wylosowania grafu E-R o e krawędziach i n wierzchołkach wynosi $\binom{\binom{n}{2}}{m} p^m (1-p)^{\binom{n}{2}-m}$.
Na tej podstawie liczba wszystkich możliwych grafów E-R o n wierzchołkach wynosi

$$\sum_{e=0}^{\binom{n}{2}} \binom{\binom{n}{2}}{e} = 2^{\binom{n}{2}} \tag{2}$$

przy czym $\binom{\binom{n}{2}}{e}$ oznacza liczbę e-elementowych kombinacji zbioru utworzonego ze wszystkich par zbioru n-elementowego.

Niestety, model taki nie jest najlepszym kandydatem do *naśladowania* obiektów rzeczywistych. Przy $p << 1$ rozkład stopni wierzchołków dany jest rozkładem Poissona, tj. rozkładem, który stosowany jest do zdarzeń rzadkich występujących w określonym przedziale czasu. Grafy generowane w tym modelu nie są w stanie dobrze odwzorowywać *hub*-ów, tj. skupisk.
Modele, które są wolne powyżej opisanych wad grafów opartych o model E-R, oparte są o rozkłady potęgowe i zostaną opisane w następnej sekcji (3).

4 Model Barabasiego-Albert

Jak się okazało, do modeli sieci rzeczywistych, np. sieci WWW czy sieci społecznościowych nie bardzo pasuje model E-R. W kontekście sieci WWW(gdzie *połączenie* między węzłami A i B jest zdefiniowane jako umieszczenie hipertextowego linka na stronie A do strony B) wynika to z tego, że autorzy tych stron z większym prawdopodobieństwem umieszczają linki na strony bardziej popularne, niż na te mniej popularne. Czyli, strony popularne stają się jeszcze bardziej popularniejsze, czyli "rich getting richer and poor poorer". Podobne rozumowanie odpowiada siecią społecznościowym czy połączenią handlowym. Odpowiada to tzw. sieciom bezskalowym. [1]

4.1 Sieć bezskalowa

Siecią bezskalową nazywamy sieć, rozkład liczby połączeń między węzłami jest wykładniczy, czyli spełnia równanie

$$P(k) \sim k^{-\gamma} \tag{3}$$

gdzie γ nazywana jest wartością właściwą grafu i zwykle mieści się w przedziale $(2, 3)$ Oznacza to, że w sieci będziemy mieli dużo wierzchołków o małym stopniu i małą(w proporcji) liczbę wierzchołków o dużym stopniu.

4.2 Preferencyjne dołączanie wierzchołków

Mechanizm preferencyjnego dołączania wierzchołków polega na tym, że nowy wierzchołek z większym prawdopodobieństwem zostanie dołączony do starszego wierzchołka z większym stopniem:

$$P(k_i) = \frac{k_i}{\sum_{j=1}^{n} k_j} \tag{4}$$

Jest to tak zwana liniowa reguła preferencyjnego dołączania. W kontekście generatora sieci BA, w kolejnych krokach czasowych $t = 1, 2, 3$ przy dołączaniu stałej ilości wierzchołków m, prawdopodobieństwo będzie wyglądało następująco:

$$P(k_i) = \frac{k_i}{\sum_{i=1}^{t} k_i} = \frac{k_i}{2mt} \tag{5}$$

gdzie m - parametr sieci, określający ile wierzchołków dołączamy w kolejnej chwili czasowej.

4.3 Rozkład stopni wierzchołków

Celem pracy jest napisanie generatora grafów modelu BA. Jednak napisany generator musi być zweryfikowany, czyli musimy stwierdzić że wygenerowane sieci rzeczywiście należą do modelu BA. Zrobimy to na podstawie weryfikacji rozkładu stopni wierzchołków grafów. Żeby to zrobić jednak musimy wyznaczyć teoretyczny rozkład wierzchołków. Można stosować do tego dwie metody:

4.3.1 Model czasu ciągłego

Metoda czasu ciągłego polega na założeniu ciągłości czasu oraz na przyjęciu że stopnie wierzchołków $k_i(t)$ zmieniają się w czasie w sposób ciągły na ułamek stopnia(dlatego w tej metodzie posługujemy się stopniami/rozkładami uśrednionymi). Zatem w pojedynczym kroku czasowym, stopień wierzchołka k zmienia się o

$$\Delta k_i/\Delta t \simeq \partial k_i/\partial t \tag{6}$$

W tym modelu w jednej chwili czasowej węzeł może uzyskać więcej niż jedno połączenie. Rozkład prawdopodobieństwa uzyskania l połączeń jest taki sam jak w schemacie Bernoullego:

$$p(l;m) = \binom{m}{l} P(k_i)^l (1 - P(k_i))^{m-l} \tag{7}$$

Wartość oczekiwana rozkładu k wynosi np, a wariancja $\sigma^2 = np(1-p)$
Możemy zatem stwierdzić, że stopień zmiany wierzchołka wynosi

$$\frac{\partial k_i}{\partial t} = \sum_{l=0}^{m} lp(l;m) = \frac{k_i}{2t} \tag{8}$$

Możemy rozwiązać to równanie różniczkowe, mając warunek początkowy $k_i t_i = m$, gdzie t_i - moment czasowy kiedy węzeł i został dodany do sieci. W wyniku otrzymujemy, że

$$k_i(t) = m\sqrt{\frac{t}{t_i}} \tag{9}$$

Rozkład stopnia wierzchołków można znaleźć poprzez znalezienie funkcji gęstości :

$$P(k_i(t) > k) = P(t_i < \frac{mt}{k^2}) = 1 - P(t_i > \frac{mt}{k^2}) \tag{10}$$

gdzie $P(t_i) = \frac{1}{m_0+t}$ Czyli

$$P(k_i > k) = 1 - \frac{mt}{k^2}\frac{1}{m_0+t} \tag{11}$$

Różniczkując $\frac{d}{dk}P(k_i < k)$ dostajemy, że rozkład stopni wierzchołków ma następującą postać:

$$P(k_i = k) = \frac{2m^2 t}{k^3}\frac{1}{m_0+t} \sim \frac{2m^2}{k^3} \tag{12}$$

4.3.2 Model równania *master*

Przy wykorzystaniu równania *master* wyznaczamy ścisły rozkład prawdopodobieństwa wierzchołków, dlatego że operujemy na stopniach węzłów a nie średnich stopni węzłów.
W tej metodzie posługujemy się równaniem mistrza, które opisuję zmianę w czasie ciągłym stanu układu fizycznego(w naszym przypadku - grafa). Chcemy uzyskać zatem równanie różniczkowe pierwszego rzędu, opisujące zmianę w czasie prawdopodobieństwa P_i znalezienia układu w stanie i, przy tym że w każdej chwili czasowej układ może zmienić swój stan i na dowolny j: $i \to j$. Zakładamy, że tempo takich zmian jest $T_{i\to j}$:

$$\frac{dP_i}{dt} = \sum_j P_j T_{j\to i} - \sum_j P_i T_{i\to j} \tag{13}$$

Wyznaczmy zatem $p_i(k, t_i, t)$ jako prawdopodobieństwo że w chwili t węzeł i dodany w chwili t_i ma stopień k:

$$p_i(k, t_i, t+1) = \sum_{l=0}^{m} p(l; m) p_i(k-l, t_i, t) \tag{14}$$

Żeby uzyskać rozkład dla wszystkich wierzchołków musimy zsumować te pw-wa p_i. Uzyskujemy

$$P(k, t) = \frac{1}{t} \sum_{t_i=1}^{t} p_i(k, t_i, t). \tag{15}$$

szukając różniczki uzyskujemy że

$$p_i(k, t_i, t+1) = \frac{k-1}{2t} p(k-1, t_i, t) + (1 - \frac{k}{2t}) p(k, t_i, t). \tag{16}$$

Rozkład wierzchołków można uzyskać

$$P(k) = \lim t \to \infty P(k, t) \tag{17}$$

Co daje nam równanie rekurencyjne

$$P(k) = \frac{k-1}{k-2} P(k-1) \tag{18}$$

Gdzie warunek początkowy $P(m) = \frac{2}{m+2}$ Rozwiązując równanie rekurencyjne dostajemy rozkład wierzchołków w sieci BA:

$$P(k) = \frac{2m(m+1)}{k(k+1)(k+2)} \tag{19}$$

przy czym $k \geqslant m$ Zauważmy różnicę rozkładów, którą jednak da się wytłumaczyć tym że równanie master jest metodą ścisłą, gdzie przybliżenia wynikały z poszukiwaniem granicznego rozkładu prawdopodobieństwa $P(k)$ dla $t \to \infty$

5 Zastosowana technologia oraz stuktury danych

5.1 Technologia

Do stworzeia rozwiązania zostaną użyte dwa języki programowania:

- C++
- R

W języku C++ przy użyciu biblioteki `STL` zostanie zaimplementowany generyczny model grafu oraz niezbędne metody pozwalające:

- zainicjować pusty graf w postaci **listy sąsiedztwa**
- stworzyć wierzchołek
- sprawdzić stopnie wierzchołków
- stworzyć krawędź między wierzchołkami
- dodać krawędź do grafu

- wygenerować graf losowy w modelu BA
- zapisać graf z pamięci programu do formatu z rodziny `XML`

Język R oraz związany z nim ekosystem `tidyverse` zostanie użyty do stworzenia wizualizacji wyników eksperymentów numerycznych. Raport z eksperymentów zostanie przedstawiony w postaci interaktywnego notatnika `Jupyter notebook`.

5.2 Architektura oprogramowania

6 Eskperymenty numeryczne

Po wygenerowaniu grafu zostanie zbadany rozkład stopni wierzchołków. W związku z tym, że tworzony graf jest losowy dla ustalonych parametrów modelu BA, tj. (n, n_0, m) niezbędne będzie wykonanie serii powtórzonych eksperymentów oraz uśrednienie wyników. Działanie takie pozwoli uzyskać próbę wiarygodną statystycznie.
W celu porównania zgodności empirycznych rozkładów stopni wierzchołków z rozkładami teoretycznymi zostaną wykonane następujące czynności:

1. sprawdzona zostanie hipoteza statystyczna zgodności rozkładów (np. test Kołmogorowa-Smirnowa lub testy oparte na $\mathcal{E}$-statystyce)
2. policzone i porównane zostanę dywergencje Kullbacka-Leiblera dla rozkładu P_1 oraz P_2

$$\Delta(P_1, P_2) = \sum_i P_1(i) log \frac{P_1(i)}{P_2(i)} \tag{20}$$

3. zostanie stworzony jądrowy estymator funkcji gęstości prawdopodobieństwa i zwizualizowany na tle teoretycznego rozkładu gęstości prawdopodobieństwa.

Metaparametr eksperymentu $L_{(m,n,n_0)}$, tj. liczbę powtórzeń eksperymentu dla zadanych wartości parametrów modelu BA, na chwilę obecną jest proponowany jako 20. Metaparametr ten będzie służył do indeksowania wyników eksperymentów numerycznych, które w zależności od wykonywanej procedury będą:

- tabelaryczne (hipotezy statystyczne)
- graficzne (dywergencje oraz histogramy)

Sprawozdanie 3
Wnioski końcowe

Literatura

[1] Model barabasiego-albert (ba). http://www.if.pw.edu.pl/~agatka/moodle/modele_SE_BA.html.

[2] P. Erdős and A Rényi. On the evolution of random graphs. In *PUBLICATION OF THE MATHEMATICAL INSTITUTE OF THE HUNGARIAN ACADEMY OF SCIENCES*, pages 17–61, 1960.

[3] Agata Fronczak. Wykładnicze grafy przypadkowe: teoria, przykłady, symulacje numeryczne. 2014.

[4] Ali Hadian, Sadegh Nobari, Behrooz Minaei-Bidgoli, and Qiang Qu. Roll: Fast in-memory generation of gigantic scale-free networks. In *Proceedings of the 2016 International Conference on Management of Data*, SIGMOD '16, pages 1835–1838, New York, NY, USA, 2016. ACM.

[5] Ali Hadian, Sadegh Nobari, Behrooz Minaei-Bidgoli, and Qiang Qu. Roll: Fast in-memory generation of gigantic scale-free networks. In *Proceedings of the 2016 International Conference on Management of Data*, SIGMOD '16, pages 1829–1842, New York, NY, USA, 2016. ACM.