Zastosowanie szkiców danych w analizie dużych grafów

Paweł Polerowicz

Praca napisana pod kierunkiem **dr. inż. Jakuba Lemiesza**

24 czerwca 2024

Główne sposoby modelowania problemu

- Model klasyczny
- Model strumieniowy
- Model półstrumieniowy
- Model rozproszony

Definicja 1 (Strumień grafowy)

Strumieniem grafowym nazywamy ciągłą sekwencję elementów, z których każdy ma postać trójki:

$$e_i = (< s_i, d_i >; w_i, t_i),$$

gdzie s_i, d_i wierzchołki grafu i przez parę $< s_i, d_i >$ oznaczamy krawędź pomiędzy nimi. Z kolei w_i i t_i to odpowiednio waga tej krawędzi i moment jej wystąpienia.

Cele pracy

- Dokonanie przeglądu istniejących rozwiązań w dziedzinie analizy strumieni grafowych.
- Szczegółowe omówienie metod generujących zanurzenia wierzchołków grafu, czyli ich reprezentacje w niskowymiarowej przestrzeni wektorowej.
- Użycie schematu szkicowania opartego na próbkach z rozkładu wykładniczego do tworzenia zanurzeń wierzchołków.
- Wykorzystanie operacji teoriomnogościowych do efektywnego operowania na szkicach danych.
- Sprawdzenie skuteczności rozwiązania poprzez analizę precyzji rekonstrukcji krawędzi w grafie oraz dokonanie analizy złożoności obliczeniowej.

Zanurzenia wierzchołków

Definicja 2

Zanurzeniem wierzchołka v nazywamy wektor $\overline{S}^v \in \mathbb{R}_+^m$, *wyznaczany na podstawie cech danego wierzchołka. Będziemy przyjmować* $m \ll |V|$.

- Zanurzenia wierzchołków stanowią efektywną pamięciowo reprezentację grafu – złożoność $O(m|V|)$.
- Mogą być tworzone na różne sposoby, np. próbkowanie wierzchołków i spacery losowe, faktoryzacja macierzy sąsiedztwa, **wykorzystanie funkcji haszujących do generowania próbek z rozkładu wykładniczego**.
- Taka reprezentacja może zostać łatwo wykorzystana jako dane w uczeniu maszynowym.

NodeSketch (Yang, Rosso, Li, Cudre-Mauroux, 2019)

Definicja 3 (k-sąsiedztwo)

k-sąsiedztwem wierzchołka v w grafie $G = (V, E)$ nazywamy maksymalny podzbiór V taki, że między v, a każdym z tworzących go wierzchołków istnieje ścieżka długości co najwyżej $k - 1$.

Algorytm `NodeSketch` wykorzystuje m niezależnych funkcji haszujących do rekurencyjnego generowania próbek z rozkładu wykładniczego na podstawie ich k-sąsiedztwa.
Schemat obliczania j-tej próbki dla wektora $V = [V_1, V_2, \ldots, V_D]$:

$$S_j = \underset{i \in \{1,2,\ldots,D\}}{\arg\min} \frac{-\log h_j(i)}{V_i}.$$

ExpSketch (Lemiesz, 2021)

Metoda generowania próbek dla krawędzi i o wadze λ_i w algorytmie `ExpSketch`:

$$E = -\frac{\ln(h(i||k))}{\lambda_i} \sim Exp(\lambda_i).$$

Twierdzenie 1 (Suma szkiców)

Niech A, B - szkice zbiórów $\mathbb{A}, \mathbb{B}$. Wtedy szkic ich sumy możemy wyznaczyć (z własności rozkładu wykładniczego) jako:

$$A \cup B = (\min\{A_1, B_1\}, \min\{A_2, B_2\}, \ldots, \min\{A_m, B_m\}).$$

ExpSketch (Lemiesz, 2021)

Definicja 4 (Ważone podobieństwo Jaccarda)

Ważone podobieństwo Jaccarda to miara podobieństwa dwóch zbiorów, zdefiniowanae jako stosunek sumy wag elementów wspólnych do sumy wag elementów w sumie mnogościowej zbiorów.

$$J_w(\mathbb{A}, \mathbb{B}) = \frac{|\mathbb{A} \cap \mathbb{B}|_w}{|\mathbb{A} \cup \mathbb{B}|_w}.$$

Twierdzenie 2

Nieobciążony estymator ważonego podobieństwa Jaccarda zbiorów $\mathbb{A}$ i $\mathbb{B}$ można otrzymać, wykorzystując ich szkice:

$$\hat{J}_w(A, B) = \frac{1}{m} \sum_{k=1}^{m} 1[A_k = B_k].$$

EdgeSketch

Algorithm EdgeSketch($\tilde{A}, m$)

foreach rząd r w $\tilde{A}$ **do**
  $ns \leftarrow [\,]$; `// lista sąsiadów`
  foreach $i \in \{1, \ldots, |V|\}$ **do**
    if $\tilde{A}[r, i] \neq 0$ **then**
      $ns \leftarrow ns \cup \{((\min(i, r)||\max(i, r)), \tilde{A}[r, i])\}$
  $S^r \leftarrow FastExpSketch(ns, m)$
return S

Definicja 5 (Złożoność czasowa)

Złożoność czasowa `EdgeSketch` *wynosi ogółem* $O(m(|V|)^2)$, *a w średnim przypadku dla grafów nieważonych:*

$$O((|V|)^2 + |V|(m \ln(m) \ln(|V|)))$$

EdgeSketch

- Skuteczność algorytmów została zmierzona poprzez precyzję rekonstrukcji krawędzi w grafie.
- Prawdopodobieństwo wystąpienia krawędzi jest oceniane na podstawie podobieństwa Jaccarda między wierzchołkami – w eksperymentach obliczano macierz podobieństwa Jaccarda na podstawie zanurzeń i wybierano z niej t najwyższych wartości.
- W algorytmie `NodeSketch` jest ona obliczana raz, na podstawie zanurzeń wygenerowanych dla danego k.
- Ostateczna macierz podobieństwa w algorytmie `EdgeSketch` powstaje przez połączenie macierzy niższych rzędów z odpowiednimi wagami:

$$simM = \sum_{k=2}^{K} \alpha^{k-2} simM_k.$$

Wpływ rozmiaru szkicu na precyzję rekonstrukcji

		NodeSketch				EdgeSketch			
b	**k**	$\mathbf{t = 100}$	$\mathbf{t = 1000}$	$\mathbf{t = 10000}$	$\mathbf{t = \lvert E \rvert}$	$\mathbf{t = 100}$	$\mathbf{t = 1000}$	$\mathbf{t = 10000}$	$\mathbf{t = \lvert E \rvert}$
2	2	0.57	0.525	0.5136	0.5072	1	1	0.4391	0.2616
	3	0.47	0.527	0.5089	0.5016	1	1	0.4545	0.3426
	4	0.44	0.523	0.5079	0.5022	1	1	0.4545	0.3426
4	2	0.5	0.542	0.5343	0.5131	1	1	0.3352	0.1547
	3	0.53	0.483	0.4991	0.5003	1	1	0.5342	0.4087
	4	0.46	0.499	0.4983	0.5008	1	1	0.5342	0.4087
8	2	0.66	0.594	0.5521	0.5234	1	1	0.2831	0.1289
	3	0.51	0.528	0.5158	0.5063	1	0.884	0.5825	0.4762
	4	0.5	0.496	0.5002	0.5029	1	0.884	0.5821	0.4755

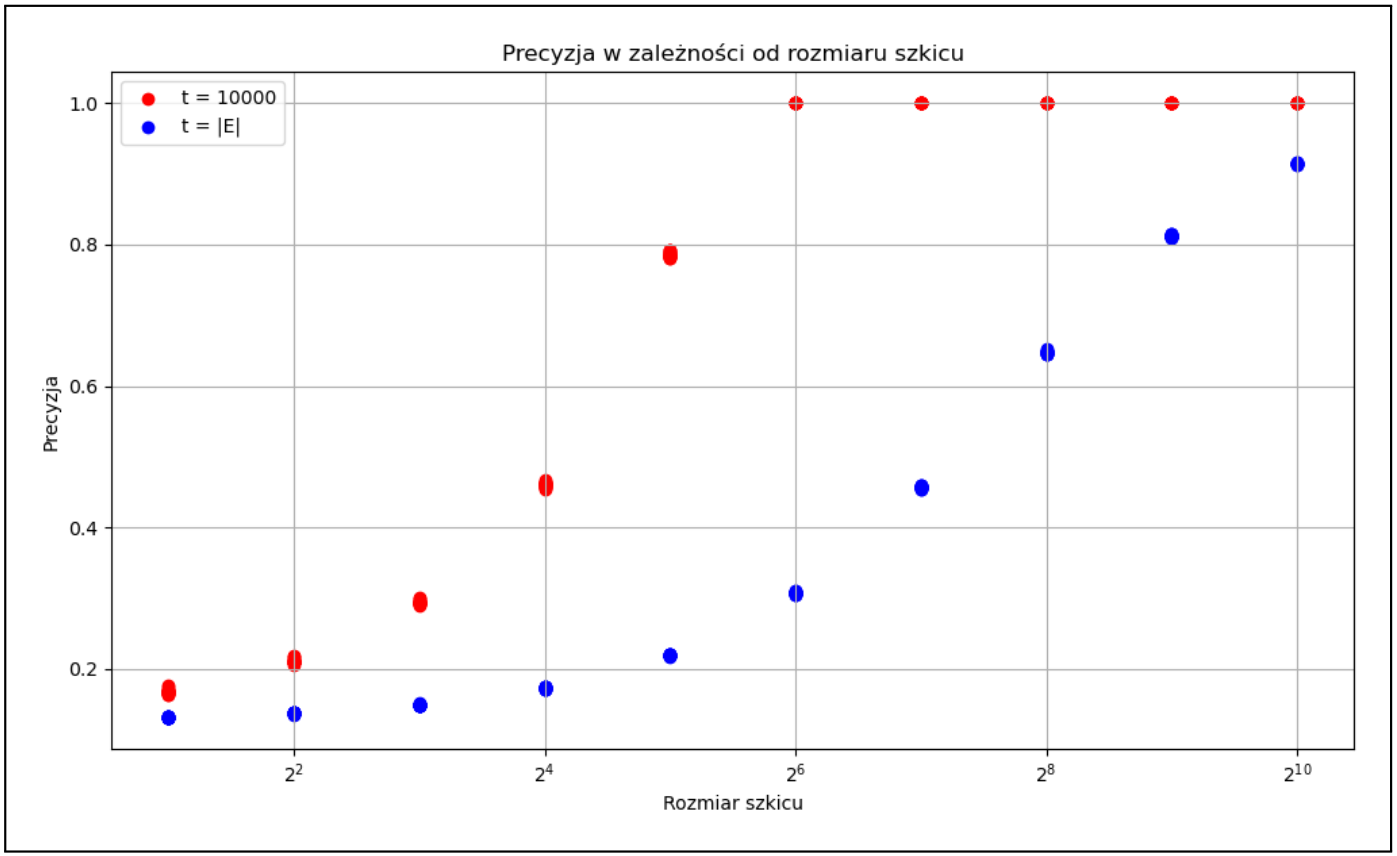
Tabela: Precyzja uzyskiwana przez algorytmy NodeSketch i EdgeSketch dla grafów w stochastycznym modelu blokowym dla różnych liczb bloków b oraz wielkości próbek t.

Wpływ rozmiaru szkicu na precyzję rekonstrukcji

		NodeSketch				EdgeSketch			
p	**k**	**t = 100**	**t = 1000**	**t = 10000**	**t = |E|**	**t = 100**	**t = 1000**	**t = 10000**	**t = |E|**
0.0005	2	0	0	0.0012	0.0031	1	1	1	0.5505
	3	0	0	0.007	0.0248	1	1	0.94	0.6125
	4	0	0.006	0.0122	0.012	1	1	0.9207	0.6073
0.001	2	0	0	0.0016	0.0055	1	1	1	0.3721
	3	0.01	0.017	0.012	0.006	1	1	0.9671	0.4235
	4	0	0.003	0.0084	0.0099	1	1	0.9714	0.4275
0.005	2	0.01	0.008	0.0054	0.0065	1	1	1	0.0989
	3	0	0.007	0.0086	0.0063	1	0.941	0.3344	0.0581
	4	0	0.004	0.0064	0.0088	1	0.941	0.3344	0.0581
0.01	2	0	0.01	0.0086	0.0107	1	1	1	0.0583
	3	0.03	0.012	0.0101	0.0117	1	1	1	0.05
	4	0.01	0.011	0.0151	0.0138	1	1	1	0.05

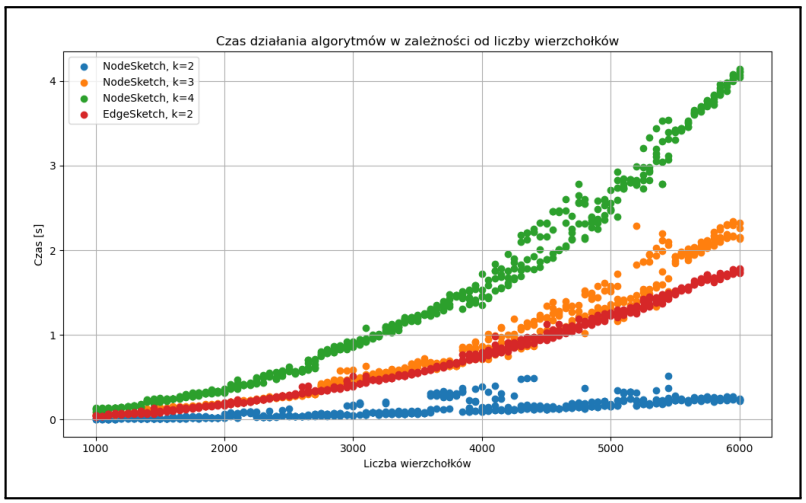
Tabela: Precyzja uzyskiwana przez algorytmy NodeSketch i EdgeSketch dla grafów ważonych w modelu Erdosa-Renyiego.

Wpływ rozmiaru szkicu na precyzję rekonstrukcji

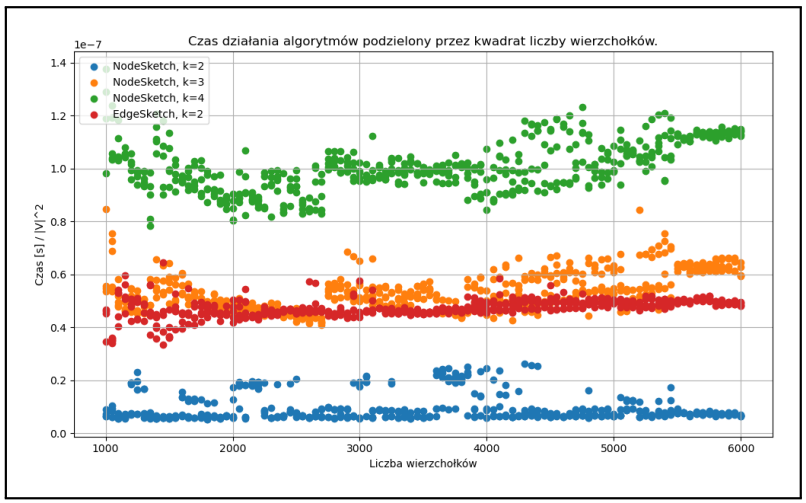


Rysunek: Precyzja uzyskiwana przez algorytm EdgeSketch w zależności od rozmiaru szkicu dla grafu w stochastycznym modelu blokowym.

Złożoność obliczeniowa



Rysunek: Czas działania algorytmów w zależności od liczby wierzchołków.



Rysunek: Czas działania algorytmów podzielony przez kwadrat liczby wierzchołków.

Dziękuję za uwagę.