Metody probabilistyczne Algorytmiki

Jacek Cichoń

Politechnika Wrocławska

Seminarium dyscypliny informatyka techniczna i telekomunikacja

Propagacja ekstremów - I

Model

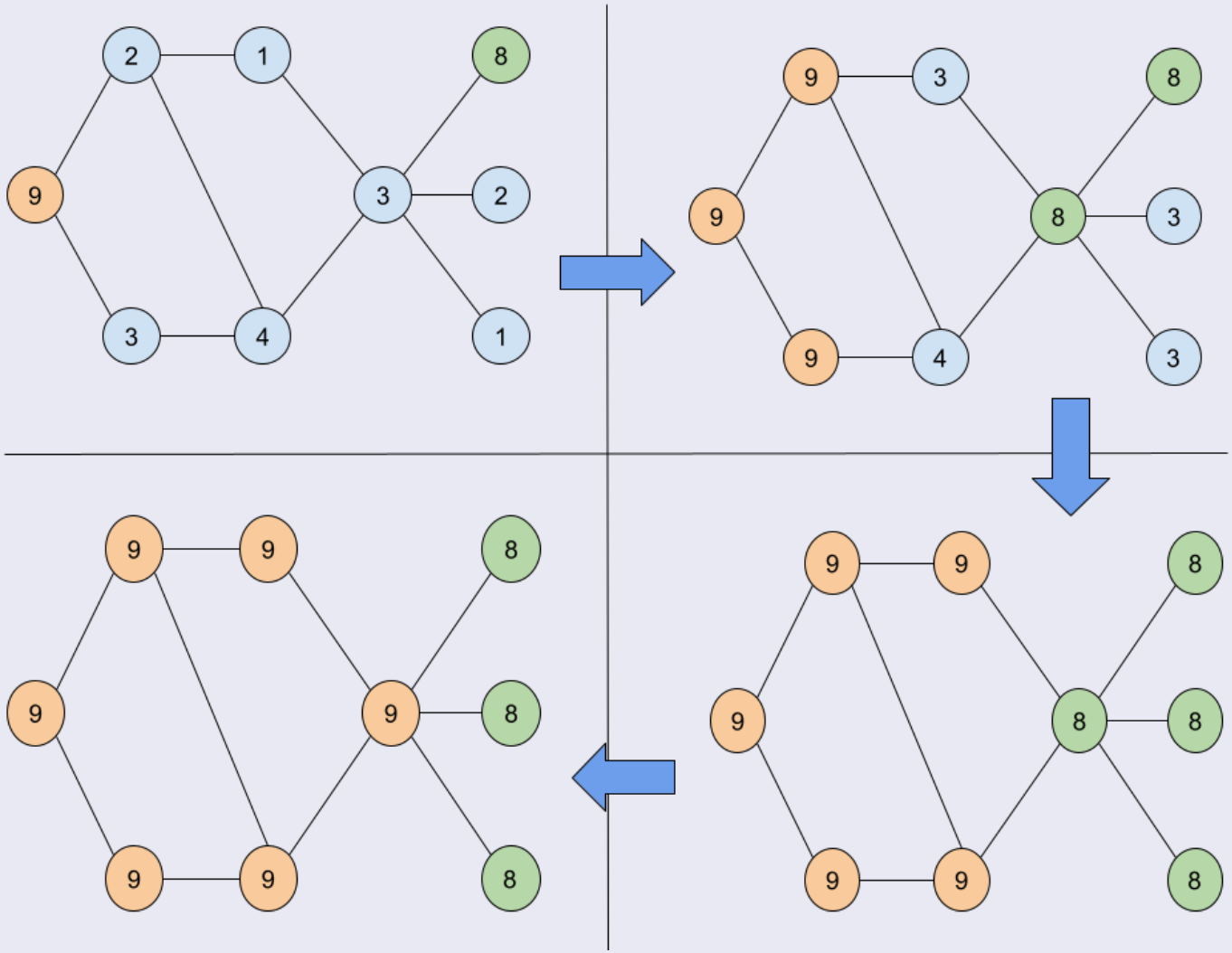
- $\mathcal{G} = (V, E)$: skończony spójny graf prosty
- $T : V \to \mathbb{R}$: obserwowane parametry przez wierzchołki
- CEL: wyznacz $\max\{T(v) : v \in V\}$ i rozpropaguj tę wartość między wszystkie wierzchołki

Historia

- Carlos Baquero, Paulo Sergio Almeida, Raquel Menezes, and Paulo Jesus, *Extrema Propagation: Fast Distributed Estimation of Sums and Network Sizes*, IEEE Trans. Parallel Distrib. Syst.,2012
- ...

Propagacja ekstremów - II

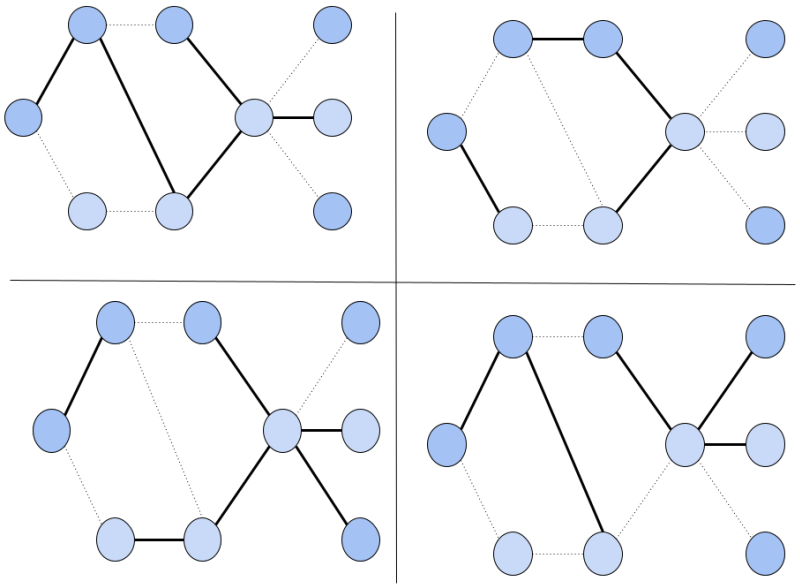
Podstawowy algorytm



Propagacja ekstremów - III

Model transmisji z błędami

- $X : E \times \mathbb{N} \to \{0, 1\}$
- $\Pr[X(e, t)] = p$, gdzie $p \in (0, 1]$ (prawdopodobieństwo sukcesu)
- Rodzina $\{X(e, t) : e \in E \wedge t \in \mathbb{N}\}$ jest niezależna



Propagacja ekstremów - IV

Twierdzenie (J. Cichoń, D. Dworzański, K. Gotfryd)

Niech $\mathcal{G} = (V, E)$ *będzie grafem spójnym,* $|V| = n + 1$, $n > 0$. *Niech* D *oznacza średnicę grafu* $\mathcal{G}$. *Niech* $a \in V$. *Wtedy*

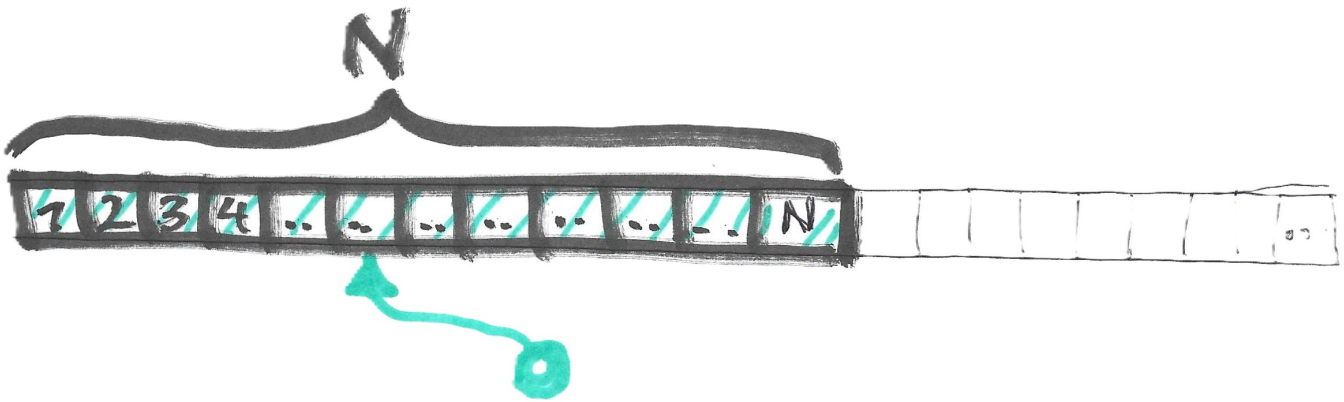
1. $\mathrm{E}[T(\mathcal{G}, \{a\}, p)] \leqslant \frac{D \ln \frac{ne}{D}}{\ln \frac{1}{1-p}} + D$
2. *Dla każdego* $\varepsilon \in (0, 1)$ *mamy*

$$\Pr\left(T(\mathcal{G}, \{a\}, p) \leqslant \frac{D \ln\left(\frac{n}{\varepsilon}\right)}{\ln\left(\frac{1}{1-p}\right)} \right) \geqslant 1 - \varepsilon$$

Praca: J. Cichoń, D. Dworzański, K. Gotfryd

On Reliability of Extrema Propagation Technic in Random Environment, konferencja: OPDIS 2024, 11-13 grudnia 2024, Lucca, Włochy

Probkowanie strumienia danych

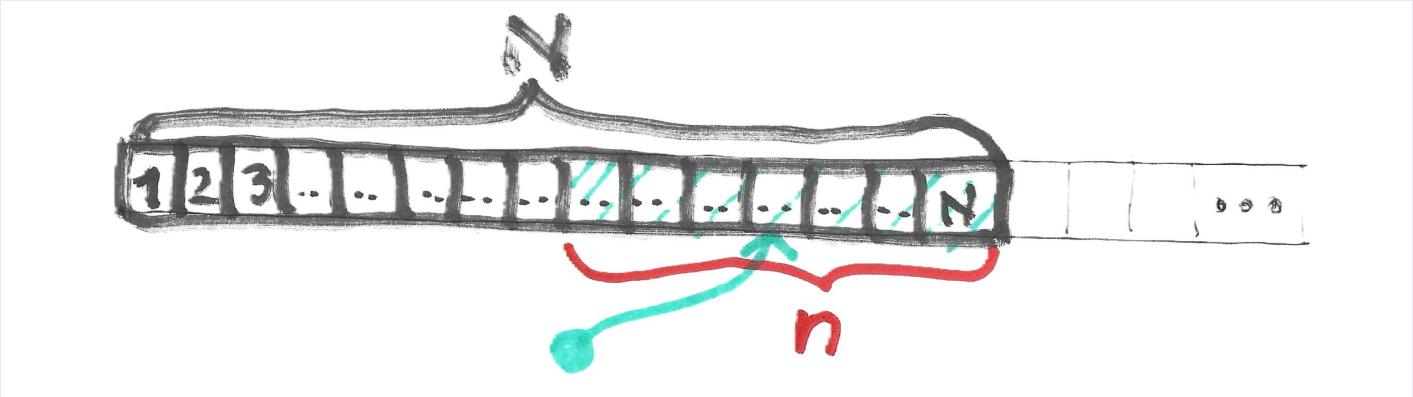


Vitter, 1985: podstawowa wersja

```
INIT: n=0; L = 0; ptr=NIL;
onGet(X: stream){
  n++;
  if (Random() < 1/n){
   L = n;
  ptr = X
 }
}
```

Sliding window - I

Opis problemu



1. Obserwujemy strumień danych $[x_1, x_2, \ldots, x_{t-1}, x_N, \ldots]$
2. Mamy ustaloną liczbę n
3. Chcemy wylosować element x_i z ciągu $[x_{N-n+1}, x_{N-n+2}, \ldots, x_{N-1}, x_N]$

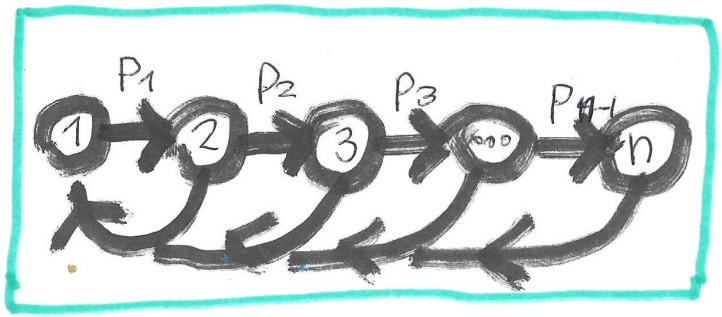
Sliding window - II

Historia

1. **SODA 2002**: Brian Babcock, Mayur Datar, Rajeev Motwani (średnia złożoność pamięciowa: e; pesymistyczna złożoność pamięciowa: $O(n)$)
2. **The Journal of Computer and System Sciences, 2012**: Vladimir Braverman, Rafail Ostrovsky, Carlo Zaniolo (złożoność pamięciowa: $O(1)$; dwa wskaźniki !!!)
3. wszystkie rozwiązania dla rozkładu jednostajnego

Motwani (SODA 2002): znalezionemu algorytmowi wiele brakuje do elegancji algorytmu Vittera

Sliding window - II



1. rozważany łańcuch Markowa jest ergodyczny; ma więc rozkład stacjonarny π
2. PYTANIE: jakie rozkłady możemy otrzymać?
3. Czy możemy otrzymać rozkład jednostajny?

Sliding window - III

Twierdzenie

Rozkład prawdopodobieństwa π na $\{1, \ldots, n\}$ jest rozkładem stacjonarnym właściwych diabelskich schodów wtedy i tylko wtedy, gdy

$$\frac{F[i-1] + F[i+1]}{2} < F[i]$$

dla wszystkich $i \in \{2, \ldots, n-1\}$, gdzie $F[i] = \sum_{k=1}^{i} \pi_i$.

1. Rozkład jednostajny na $\{1, \ldots, n\}$ nie ma tej własności !!! :-(

Sliding window - III

Twierdzenie

Rozkład prawdopodobieństwa π na $\{1, \ldots, n\}$ jest rozkładem stacjonarnym właściwych diabelskich schodów wtedy i tylko wtedy, gdy

$$\frac{F[i-1] + F[i+1]}{2} < F[i]$$

dla wszystkich $i \in \{2, \ldots, n-1\}$, gdzie $F[i] = \sum_{k=1}^{i} \pi_i$.

1. Rozkład jednostajny na $\{1, \ldots, n\}$ nie ma tej własności !!! :-(
2. D. Bojko: rozkład $F(k) = \sqrt{\frac{k}{n}}$ ma tą własność oraz jeśli X, Y są niezależne i mają rozkład o tej dystrybuancie to zmienna $\max\{X, Y\}$ ma rozkład jednostajny !!!

Sliding window - IV

Odtworzenie prawdopodobieństw przejść

$$p_i = \frac{\pi_i}{\pi_{i+1}} = \frac{F[i] - F[i-1]}{F[i+1] - F[i]} = \ldots = 1 - \frac{\sqrt{i+1} + \sqrt{i-1}}{\sqrt{i+1} + \sqrt{i}}$$

1. Kolejny problem: jak szybka jest zbieżność do rozkładu stacjonarnego?

Sliding window - IV

Odtworzenie prawdopodobieństw przejść

$$p_i = \frac{\pi_i}{\pi_{i+1}} = \frac{F[i] - F[i-1]}{F[i+1] - F[i]} = \ldots = 1 - \frac{\sqrt{i+1} + \sqrt{i-1}}{\sqrt{i+1} + \sqrt{i}}$$

1. Kolejny problem: jak szybka jest zbieżność do rozkładu stacjonarnego?
2. Pomysł: wkocz od razu do rozkładu stacjonarnego

Sliding window - IV

Odtworzenie prawdopodobieństw przejść

$$p_i = \frac{\pi_i}{\pi_{i+1}} = \frac{F[i] - F[i-1]}{F[i+1] - F[i]} = \ldots = 1 - \frac{\sqrt{i+1} + \sqrt{i-1}}{\sqrt{i+1} + \sqrt{i}}$$

1. Kolejny problem: jak szybka jest zbieżność do rozkładu stacjonarnego?
2. Pomysł: wkocz od razu do rozkładu stacjonarnego
3. INIT: $X = \left\lceil n \cdot \mathrm{random}()^2 \right\rceil$ (inverse transform method)

Sliding window - V

Algorytm dla rozkładu jednostajnego

1. użyj dwóch niezależnych kopii X, Y diabelskich schodów dla $F(k) = \sqrt{\frac{k}{n}}$
2. w momencie żądania odczytu zwróć to co zapamiętał większy z licznikow X, Y

Naszą metodą można generować dużą klasę rozkładów.

Praca: D. Bojko, J. Cichoń, M. Kutyłowski

Sliding Window Sampling Over Streaming Data. A solution based on devil's Markov chains, DSAA 2023, Thessaloniki, Grecja, 9-13 października, 2023

KLASYKA: licznik Morrisa

Fakt z I roku studiow

Do zaprezentowania dowolnej liczby z zakresu $0, \ldots, n$ potrzeba

$$\lceil \log_2(n+1) \rceil$$

bitów.

Klasyczny licznik

```
INIT:  C=0

onTick() {
   C++
}
```

Licznik Morrisa

```
INIT:  C=0

onTick() {
  if (Random()< (1/2)^C) {
   C++
  }
}
```

Robert Morris, 1977, Bell Labs

Licznik Morrisa

Twierdzenie

Niech C_n oznacza wartość licznika Morrisa po n wywołaniach procedury `onTick()`. Wtedy

1. $\mathrm{E}[2^{C_n}] = n + 1$
2. $\mathrm{var}[2^{C_n}] = \frac{1}{2}n(n+1)$

Wniosek

Liczba $2^C - 1$ jest nieobciążonym estymatorem liczby wywołań licznika Morrisa.

Wniosek

Wniosek

1. $C_n \approx \log_2(n+1)$
2. do zapamiętania liczby L_n potrzeba w przybliżeniu $\log_2(\log_2(n))$ bitów

Licznik Morrisa

Twierdzenie (Philippe Flajolet, 1985)

$$\mathrm{E}[C_n] = \log_2(n) - c + \omega(\log_2(n)) + \mathrm{O}(n^{-0.98}) \ ,$$

gdzie $c = 0.2739...$ *oraz* ω *jest funkcją okresową o okresie 1 taką, że* $|\omega(x)| < 10^{-5}$ *dla każdego* x.

$$\log_2(\log_2(2^{64})) = \log_2(64) = 6$$

To jest koniec

Dziękuję