Потоковые алгоритмы на графах

Всеволод Опарин

CS Клуб, Осень 2014

23 ноября

Спарсификаторы

- $G = \langle V, E, w \rangle$.
- Разрез $\delta(U) = \{uv \mid u \in U, v \notin U\}$.
- Спаннер сохраняет расстояние, спарсификатор – разрез.

Определение

Взвешенный подграф H является $(1+\varepsilon)$-спарсификатором невзвешенного графа G, если

$$\forall U \subseteq V \quad w(\delta_G(U)) \le w(\delta_H(U)) \le (1+\varepsilon) w(\delta_G(U)).$$

Теорема Spielman, Teng. Оффлайн

По произвольному графу можно построить $(1+\varepsilon)$-спарсификатор с $\widetilde{O}(n\varepsilon^{-2})$ ребрами за $\widetilde{O}(m)$.

Спарсификаторы. Потоковый режим

Лемма 1

H_1, H_2 – α-спарсификаторы для G_1, G_2. Тогда $H_1 \cup H_2$ – α-спарсификатор для $G_1 \cup G_2$.

Лемма 2

L – α-спарсификатор для H, H – α-спарсификатор для G. Тогда L – α^2-спарсификатор для G.

Алгоритм:

- Делим на блоки по $O(n\delta^{-2})$ ребер.
- Пользуемся алгоритмом из Теоремы.
- Получаем $(1+\delta)^{\log m}$-спарсификатор.
- $\delta = \frac{\varepsilon}{\log m}$, чтобы получить $(1+\varepsilon)$-спарсификатор.
- Память: $O(n\varepsilon^{-2}\log^3 m)$.

Спарсификаторы. Оффлайн

Алгоритм для невзвешенного графа.

- Считаем связность для каждого ребра.
- Для каждого ребра зададим нулевой вес.
- ρ раз попробуем увеличить вес каждого ребра.
- Увеличиваем вес на $\frac{k_e}{\rho}$ с вероятностью $\frac{1}{k_e}$.

Оценка

- $\mathbf{E}\left[\, w_e \,\right] = 1$.
- $\forall S \subseteq V\ \mathbf{E}\left[\, \delta(S) \,\right] = \delta(S)$.
- Число ненулевых ребер $\leq \sum_{i=1}^{\rho} \sum_{e \in E} X_{i,e}$.
- $\mathbf{E}\left[\, \sum_i \sum_e X_{i,e} \,\right] = \rho \sum_e \frac{1}{k_e} \leq (n-1)\rho$.
- $\rho = 100 \log^3 n \varepsilon^{-2}$.

Эскиз графа

- Матрица инцидентности $M \in \{-1, 0, 1\}^{n \times \binom{n}{2}}$.

$$M_{x,(u,v)} = \begin{cases} -1, & \text{если } x = u; \\ 1, & \text{если } x = v; \\ 0, & \text{иначе.} \end{cases}$$

- b_v – ряд v. Храним для каждого линейный эскиз $A(b_v)$.
- $\delta(S) = \mathrm{supp} \sum_{v \in S} M_v$.
- l_0-сэмплинг возвращает инциндентное ребро.

l_0-сэмплирование

Задача

Храним массив `f`. Приходят запросы `f[i] += Δ`, для $i \in [n]$, $\Delta \in \mathbb{Z}$. Равновероятно выдать x из $S = \{i \mid f[i] \neq 0\}$.

Восстановление

- Строим линейный эскиз.
- Сопостовляем вектору f вектора $f_1, \cdots, f_m$.
- Для каждого вектораz будем поддерживать линейный эскиз.
- s – параметр. $h : [n] \to [n^3]$ – k-независимая.
- $f_j[i] = f[i]$, если $h(i) \leq n^3 2^{-j}$.

Число компонент связности

Алгоритм

- Изначально, в каждом блоке по вершине.
- Вытаскиваем из каждого блока по ребру через l_0-сэмплинг.
- Через СНМ мерджим множества вершин.
- $A(b_v + b_u) = A(b_v) + A(b_u)$.
- Повторить $\log m$ раз.

Проверка на k-связность

Определение

Подграф H графа G является k-сертификатом, если $\forall S \subseteq V$

$$\delta_H(S) \geq \min(\delta_G(S), k).$$

Лемма

F_i – покрывающий лес графа G без $F_1, \cdots, F_{i-1}$. Тогда $F_1 \cup \cdots \cup F_k$ – k-сертификат.

- $A(G) \to F_1$ через связность.
- Найти $B(F_1)$.
- $B(G) - B(F_1) = B(G \backslash F_1) \to F_2$ и т.д.
- Памяти $\widetilde{O}(k \cdot n)$.

Минимальный разрез

Теорема Каргера

λ – мин. разрез графа $G = \langle V, E \rangle$. Оставим каждое ребро с вероятностью
$$p \geq p^* = 6\lambda^{-1}\varepsilon^{-2}\log n.$$
и назначим вес $\frac{1}{p}$. Полученный граф – $(1+\varepsilon)$-спарсификатор G с высокой вероятностью.

- Если разрез $O(\varepsilon^{-2}\log^{O(1)} n)$, то находим тривиально.
- Иначе сэмплируем на вероятностях 2^{-i} для $i = 0 \cdots \log\frac{1}{p^*}$.
- Строим сертификаты для $k = 24\varepsilon^{-2}\log n$.
- Возвращаем разрезы, меньшие k.
- Памяти $\widetilde{O}(k \cdot n) = \widetilde{O}(\varepsilon^{-2} n)$

Спасибо!

Вопросы?