A Local Clustering Algorithm for Massive Graphs and its Application to Nearly-Linear Time Graph Partitioning

Daniel A. Spielman Shang-Hua Teng

presented by Tomáš Gavenčiak

Preliminaries

Let $G = (V, E)$ be a graph on n vertices and m edges. Let A be an adjacency matrix of G.

Let $\mu(S) = \sum_{v \in S} \deg_G(v)$. Note that $\mu(V) = 2m$.

The *conductance* of $S \subseteq V$ is defined as

$$\Phi(S) = \frac{|E(S, V \setminus S)|}{\min(\mu(S), \mu(V \setminus S))}.$$

Note that $\Phi(S) \in [0, 1]$. Let Φ_G denote the minimal $\Phi(S)$ over all noempty $S \subset V$.

Clustering problem (deciding the existence of $S \subseteq V$ with $\Phi(S) \leq \phi$ for given ϕ) is an NP-complete problem. There are $O(\sqrt{\log(n)})$ approximation algorithms, but their complexity is high (algorithms usually use maxflow, linear or semidefinite programming as subroutines).

Truncated random walks. We use several vectors $[0, 1]^V$ in the algorithm, most of these are approximations of random walk distributions, but may sum to less than 1.

Let χ_S be the characteristic $\{0, 1\}$ vector of $S \subseteq V$. Let D_S be the diagonal matrix with χ_S on the diagonal. Let $\psi_S(u) = \deg(u)/\mu(S)$ for $u \in S$ and 0 otherwise.

The random walk distribution will change according to matrix $M = (AD^{-1} + I)/2$, D is the diagonal matrix with vertex degrees. The walk stays at the current vertex with prob. 1/2 and moves across a random edge otherwise. $p_0 = \chi_v$, $p_{i+1} = Mp_i$ is the distributions of i-th step of a random walk starting from v.

For vector p, let $[p]_\epsilon(u)$ be truncated to 0 if $p(u) < \epsilon \deg(u)$, $p(u)$ otherwise. Let $q_0 = \chi_v$, $r_i = [q_i]_\epsilon$ and $q_{i+1} = Mr_i$ be the distribution of ϵ-truncated random walk after step i.

Lemma 2.2. For all vectors $p \geq 0$, $|D^{-1}(Mp)|_\infty \leq |D^{-1}p|_\infty$.

Lemma 2.4. For every $S \subseteq V$, all vectors $p, q \geq 0$ and all integers $t > 0$,

$$p^T (D_S M)^t q \leq p^T M^t q.$$

Lemma 2.5. For every $S \subseteq V$ and all integers $t > 0$,

$$\mathbf{1}^T (D_S M)^t \psi_S \geq 1 - t\Phi(S)/2.$$

Best vertices and magical function. Let $S_j(p)$ denote the set of j vertices maximizing $p(u)/d(u)$. Let $\lambda_j(p) = \mu(S_j(p))$ be the degree sum of these vertices.

For p vector and $x \in [0, 2m]$, define

$$I(p, x) = \max_{w \in [0,1]^V,\ w \cdot \deg = x} w \cdot p$$

This is a concave, non-decreasing function to $[0, 1]$ for p (sub)distribution. Note that $I(p, \lambda_j(p)) = p \cdot \chi_{S_j(p)}$ and $I(p, .)$ is linear between such points.

Let $I_x(p, x) = I(p, x)/dx$ (defined by right limit in turning points).

Algorithms

We fix $\phi \in [0, 1]$ to be the desired upper bound on conductance.

Informal statement about Nibble. For a cluster C_0 of conductance at most $f = \Omega(\phi^2/\log^3(n))$, Nibble started at random $v \in C_0$ (sampled acc. to degrees) returns C of conductance at most ϕ, mostly contained in C_0, in time linear in C with probability at least 1/2.

Important constants. Let $l = \lceil \log_2(m) \rceil$ and $t_1 = \lceil \frac{2}{\phi^2} \ln(c_1(l + 2)\sqrt{m}) \rceil$, where $c_1 \approx 200$. The paper uses t_1 up to $t_{l+1} = t_{last}$ as an alias for $t_i = it$.

Let $f = f_1(\phi) = 1/(c_2(l + 2)(l + 1)t_1)$, where $c_2 \approx 280$. For $m \geq 1000$ and the constants above $f \geq \phi^2/(2000 \log^3(m))$.

Algorithm `Nibble`(G, v, ϕ, b) for $b \in 1 \dots l$:

1. Set $\epsilon = 1/(c_3(l + 2)(l + 1)t_1 2^b)$.
2. Set $q_0 = r_0 = \chi_v$.
3. For $t = 1$ to $(l + 1)t_1$
   (a) Set $q_t = Mr_{t-1}$ and $r_t = [q_t]_\epsilon$.
   (b) If there is j such that
      i. $\Phi(S_j(q_t)) \leq \phi$ (small conductivity)
      ii. $\lambda_j(q_t) \leq (5/6)2m$ (at most 5/6 edge endpoints)
      iii. $2^b \leq \lambda_j(q_t)$ (at least 2^b edge endpoints)
      iv. $I_x(q_t, 2^b) \geq 1/(c_4(l + 2)2^b)$ (large probability mass of many vertices)

then return $C = S_j(q_t)$.

4. Return $C = \emptyset$.

Theorem N. `Nibble`(G, v, ϕ, b) can be implemented to run in time $O(2^b \log^6(m)/\phi^4)$. Also we have:

(N.1) When $C =$ `Nibble`(G, v, ϕ, b) is non-empty, $\Phi(C) \leq \phi$, $\mu(C) \leq (5/6)2m$.

(N.2) Each $S \subseteq V$ with $\mu(S) \leq (2/3)2m$ and $\Phi(S) \leq f$ has a subset S^g (of potentially good starting vertices) with $\mu(S^g) \geq \mu(S)/2$ and such that for every $v \in S^g$ with $C =$ `Nibble`(G, v, ϕ, b) non-empty, $\mu(C \cap S) \geq 2^b$.

(N.3) The set S^g may be partitioned into $S^g_0, \ldots S^g_l$ such that for every $v \in S^g$, there is b, such that if $v \in S^g_b$ then `Nibble`(G, v, ϕ, b) is non-empty.

Algorithm `RandomNibble`(G, ϕ)

1. Choose v according to ψ_V.
2. Choose $b \in 1 \ldots \lceil \log_2(m) \rceil$ with pp. proportional to 2^{-b}.
3. Return $C =$ `Nibble`(G, v, ϕ, b)

Theorem RN. The expected running time of `RandomNibble` is $O(\log^7(m)/\phi^4)$. If C ic non-empty, $\Phi(C) \leq \phi$ and $\mu(C) \leq (5/6)2m$. (N.1)

Also, for every $S \subseteq V$ with $\mu(S) \leq (2/3)2m$ and $\Phi(S) \leq f$, $\mathbf{E}[\mu(C \cap S)] \geq \mu(S)/(4\mu(V))$.

Algorithm `Partition`(G, θ, π) for $\theta, \pi \in (0, 1)$

Let $f_2(\theta) = f_1(\theta/7)/2$. Note that $f_2(\theta) \leq \phi^2/(2 \cdot 10^5 \log^3(m))$.

1. Choose $W_0 = V$, $j = 0$, $\phi = \theta/7$.
2. While $j < 12m\lceil \ln(1/\pi) \rceil$ and $\mu(W_j) \geq (3/4)2m$,
   (a) Set $j = j + 1$
   (b) Set $D_j =$ `RandomNibble`$(G[W_{j-1}], \phi)$
   (c) Set $W_j = W_{j-1} \setminus D_j$
3. Return $D = \bigcup_i D_i$

Theorem P. The expected running time of `Partition` is $O(m \ln(1/\pi) \log^7(m)/\theta^4)$, We also have

(P.1) $\mu(D) \le (7/8)\mu(V)$.

(P.2) If D is nonempty then $\Phi(D) \le \theta$.

(P.3) For any $S \subseteq V$ with $\mu(S) \le \mu(V)/2$ and $\Phi(S) \le f_2(\theta)$, with probability at least $1 - \pi$ either

> **(P.3.a)** $\mu(D) \le (1/4)2m$ or
>
> **(P.3.b)** $\mu(S \cap D) \ge \mu(S)/2$.

Analysis of `Nibble`

Step 1: Introducing S^g, proving (N.1) and (N.2).

For each $S \subseteq V$, let S^g be all v such that for all $t \le t_{last}$, $\chi_{V \setminus S}^T M^t \chi_v \le t_{last}\Phi(S)$.

Lemma 2.7 (N.1). $\mu(S^g) \le \mu(S)/2$.

Lemma 2.8 (N.2). If $\Phi(S) \le f$, $v \in S^g$ and `Nibble`(G, v, ϕ, b) is non-empty, then $\mu(C \cap S) \le 2^{b-1}$.

Step 2: Properties of $I(p, x)$, refining S^g into G_b^g.

Lemma 2.9 [LS90]. For every vector $p \ge 0$ and x, $I(Mp, x) \le I(p, x)$.

Lemma 2.10 [LS90]. For every vector $p \ge 0$, if $\Phi(S_j(Mp)) \ge \phi$, then for $x = \lambda_j(Mp)$ and $\hat{x} = \min(x, 2m - x)$,

$$I(Mp, x) \le \frac{1}{2}(I(p, x - 2\phi\hat{x}) + I(p, x + 2\phi\hat{x})).$$

For $h \in 0 \ldots l + 1$, let $x_h(v)$ be such that $I(p_{t_h}, x_h(v)) = (h + 1)/((l + 2)c_5)$.

Let $h(v) = h_v$ be $l+1$ if $x_l(v) \ge 2m/(c_6(l+2))$, otherwise $\min\{h : x_h(v) \le 2h_{h-1}(v)\}$.

$$S_0^g = \{v : \; x_{h(v)-1}(v) \in [0, 2)\}$$
$$S_b^g = \{v : \; x_{h(v)-1}(v) \in [2^b, 2^{b+1})\}$$

Lemma. $h(v)$ are well defined, S_b^g partition S^g, $x_h < h_{h+1}$.

Step 3: Truncated random walks and clustering

Lemma 2.13. For all $u \in V$, x and t,

$$p_t(u) \ge q_t(u) \ge r_t(u) \ge p_t(u) - t\epsilon \deg(u)$$
$$I(p_t, x) \ge I(q_t, x) \ge I(r_t, x) \ge I(p_t, x) - t\epsilon x$$

The hard work is done in Lemmas 2.15 and 2.17 in the paper.