Unweighted Dist Spanners Lecture

Monday, May 20, 2019 6:42 PM

Distribute Spanners

Spanners: For a graph G, a subgraph H is an α-spanner if $\forall (u,v) \in E(G)$ we have $d_H(u,v) \leq \alpha$

$\underbrace{d_H(u,v)}$ ↳ length of the shortest u to v path in H

Baswana Sen Algorithm

- Input: Graph G, integer k
- Output: A subgraph H that is a $(2k-1)$ spanner

- The idea is to gradually construct H by adding edges over iterations. In each iterations we grow and collapse clusters of small radius

- A cluster C is a subset of vertices

- A cluster C is a subset of vertices such that $G[C]$ has a spanning tree T_C rooted at a vertex $r_C \in C$ [Center of cluster C]

radius(C) $=$ height(T_C) $=$ { max distance b/w r_C and any $v \in C$ in T_C

- Set $H = (V, \emptyset)$
  ↳ no edges

- All vertices in V are marked as available

Phase 1: (k-1 iterations)

- Iteration 1:
  - every vertex is a cluster by itself

    So we have n clusters, each of radius 1

- $\mathcal{C}_1 = \{\{v\} \mid v \in V\}$ is the

- $\mathcal{C}_1 = \{\{v\} \mid v \in V\}$ is the collection of clusters after iteration 1

- Iteration $i = 2, 3, \ldots, k-1$
  - We have clusters $\mathcal{C}_{i-1}$ from the previous iteration
  - for each cluster $C \in \mathcal{C}_{i-1}$ the center $r_C \in C$ randomly decides to become inactive with Probability $= 1 - 1/n^{1/k}$
    - if $r_C \in C$ becomes inactive then the cluster C dissolves otherwise cluster C is active

      $\therefore \Pr[C \text{ is active}] = 1/n^{1/k}$
    - Consider a vertex x such that the cluster $C(x) \in \mathcal{C}_{i-1}$ dissolves

      cluster in $\mathcal{C}_{i-1}$ containing x

cluster in $\mathcal{C}_{i-1}$ containing x

- if x has a neighbor y in G such that cluster $C(y)$ is active
  - x joins $C(y)$ using edge (x,y)

  The cluster $C(y)$ grows
  (x,y) becomes part of the rooted spanning tree $T_{C(y)}$ of the grown cluster

- other every neighbor of x is
  - in a dissolved cluster
  - or has dropped out

  - All neighbors of x that are available (i.e. has not dropped out) lie in some dissolved cluster

    - Let $y_1 \cdots y_\ell \in N_G(x)$

Let $y_1, \ldots, y_\ell \in N_G(u)$
such that each y_i lies
in a distinct (dissolved) cluster

- u adds the edges
$(u, y_1), (u, y_2), \ldots, (u, y_\ell)$
to the spanner H

- Then u drops out
  - i.e. it is marked as
  dropped out

- We do the above process
for every vertex which is
available and lies in a dissolved
cluster

- Let $C_1, C_2, \ldots, C_\ell$ be the
clusters that have remained
active and possibly grown
  - Let T_{C_i} be the rooted
  spanning tree of $G[C_i]$

Ex: why? ← — we may have grown C_i and increased height(T_{C_i}) by at most 1

— We add all edges of T_{C_i} to the spanner H

— Phase 2:

— After the last ((k-1)th) iteration of Phase 1, we obtain the clusters $\mathcal{C}_{k-1}$

— For each cluster $C \in \mathcal{C}_{k-1}$

— grow C to C' as follows: for each $v \in N_G(C)$, add v to C using edge (u,v), where $u \in N_G(v) \cap C$

— This gives a rooted spanning tree T' of $G[C']$ such that $\text{height}(T') = \text{height}(T) + 1$

$height(T') = height(T) + 1$

The rooted spanning tree of $G[C]$

- Add all edges of T' to the spanner H

- Output H as the spanner

Correctness

Lemma: If $C \in \mathcal{C}_i$ then the spanning tree T_C of $G[C]$ has $height(T_C) \le i$

Proof: In iteration 1,

$\forall\, C \in \mathcal{C}_1$, $height(T_C) = 1$

In each iteration $i = 2 \cdots k-1$ if the cluster remains active and grows, $height(T_C)$ increases by at most 1.

by at most 1.

↳ why?

∴ lemma is true $\square$

Obs: At the end of iteration i, any vertex that has not yet dropped out lies in some cluster $C \in \mathcal{C}_i$ → why?

Lemma: The output H of the algo is a $(2k-1)$ spanner

Proof: Consider an edge (u,v) that is not in S

Suppose vertex v drops out at iter $i \leq k-1$ of Phase 1 and u drops out at iter i or later or never

- Since v drops out, all clusters containing a all clusters containing a neighbor of v were dissolved at iter i

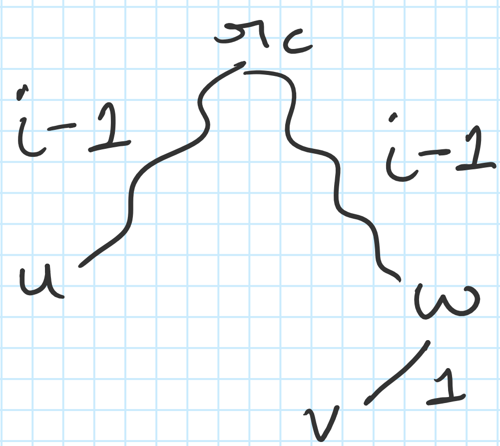
- v picks an arbitrary neighbor to each such cluster C

- let v pick $w \in C$ where C contains u

- Then $C \in \mathcal{C}_{i-1}$, which was active at iter $i-1$, added T_C to spanner H

  - height$(T_C) \leq i-1$ (above lemma)

  - $\therefore$ we have a path in H from v to u as follows



length of this path is at most

$$2(i-1)+1 \leq 2i-1$$

- The other case is when both endpoints u and v of the edge are available after iter $(k-1)$ of Phase 1

  - if u and v lie in the same cluster C of $\mathcal{C}_{k-1}$
    $\Rightarrow$ we have a path of length $2k-2$ from T_C

  - otherwise $v \in N_G(C(u))$
    $\Rightarrow v \in C'(u)$ from Phase 2
    $\therefore T'_{C(u)}$ has a path of length $2k-1$ between u and v, that is also in H

Lemma: H contains $O(k n^{1+1/k} \log n)$ edges with prob $\geq 1 - 1/n^{98}$

Pf: In phase 1, in iteration i

The clusters in $\mathcal{C}_i$ are vertex disjoint

- we add a spanning tree to H for each cluster in $\mathcal{C}_i$
  $\Rightarrow$ at most $(n-1)$ edges in total

- Some vertices drop out in this iteration, at which point they add some edges to H

Consider a vertex v that has neighbors in $c n^{1/k} \log n$ clusters of $\mathcal{C}_{i-1}$

a constant we fix to 101

Claim: $\Pr[v \text{ drops out in iteration } i] \leq \frac{1}{n^c}$

Proof: $\Pr[v \text{ drops out}]$

$= \Pr[\text{every cluster of } \mathcal{C}_{i-1} \text{ with a neighbor of } v \text{ dissolves}]$

$\leq \left(1 - \frac{1}{n^{1/k}}\right)^{c\, n^{1/k} \log n}$

$\leq (1/e)^{c \log n}$

$= 1/n^c$

Call v as HighDegree if v has neighbors in $c\, n^{1/k} \log n$ clusters of $\mathcal{C}_{i-1}$

$\Pr[\text{some high degree vertex drops out in iter } i] \leq n \times 1/n^c$

$= 1/n^{c-1}$

$= 1/n^{c-1}$

$\therefore$ with $\Pr \geq 1 - 1/n^{c-1}$ every vertex that drops out at iter i has neighbors in $\leq c\, n^{1/k} \log n$ clusters of $\mathcal{C}_{i-1}$

- so every vertex that drops out in iter i adds $c\, n^{1/k} \log n$ edges to H with $\Pr \geq 1 - 1/n^{c-1}$

  $\therefore$ total number of edges added to H by all vertices that drop out at iter i is at most $c\, n^{1+1/k} \log n$ with $\Pr \geq 1 - 1/n^{c-1}$

$\therefore$ the i-th iteration adds $O(n^{1+1/k} \log n)$ edges to H

$\therefore$ with $\Pr \geq 1 - k$

$O(n^{3/2} \log n)$ edges to H

with $\Pr \geq 1 - 1/n^{c-1}$

$\therefore \Pr$[this happens over all $k-1$ iters] (Bad Event)

$= 1 - \Pr$[Bad Event happens for some iter i]

$\leq 1 - (k-1)\, 1/n^{c-1}$ (union bound)

$\leq 1 - 1/n^{c-2}$

We also need to consider Phase 2

- Bound number of clusters in $\mathcal{C}_{k-1}$

- we start with n clusters in iter 1 (i.e. $|\mathcal{C}_1| = 1$)

- Then in each iteration i, the center of clusters

the center of clusters
that were active in iter $i-1$
randomly decide to dissolve
it's cluster, or stay active

- Consider vertex $v \in V$

Pr[v's cluster is active after
$k-1$ iterations]

$$= \left(\frac{1}{n^{1/k}}\right)^{k-1}$$

$$= 1 / n^{k-1/k}$$

$\therefore$ $\mathbb{E}$[active clusters
after $k-1$ iter] $= n \cdot 1/n^{k-1/k}$

$$= n^{1/k}$$

Pr[more than $2n^{1/k}$
active clusters after
$k-1$ iter] (Bad Event)

(Chernoff bounds)

$k-1$ (test]

$$\leq \frac{1}{e^{n^{1/k}/3}}$$

(Chernoff bounds)

$$\Pr[X \geq (1+\delta)\mu] \leq \frac{1}{e^{\frac{\delta^2 \mu}{2+\delta}}}$$

$\mu = \mathbb{E}[X]$
$= n^{1/k}$

$X = X_1 + X_2 + \cdots + X_n$

$X_i = 1$ if v_i's cluster is active

$$\leq \frac{1}{n^{100}}$$ (when n is large enough and k is const)

$$n^{100} < e^{n^{1/k}/3}$$

$\Rightarrow 300 \log n < n^{1/k}$

$\Rightarrow \log 300 + \log\log n \leq \frac{\log n}{k}$

$\therefore \Pr[\text{at most } 2n^{1/k} \text{ clusters in } \mathcal{C}_{k-1}]$
$\geq 1 - \frac{1}{n^{100}}$

$/ n^{100}$

For each of these $2n^{1/k}$ clusters
we add upto n new edges
$\left(C \in \mathcal{C}_{k-1} \text{ grows to } C' \subseteq C \cup N_G(C) \right)$

$\therefore$ in total we add
$2n^{1+1/k}$ new edges to H
in Phase 2

- In total over Phase 1 and 2
  we add $O(k n^{1+1/k} \log n)$ edges
  to the spanner H

- Now to bound total error probability

$\Pr[\text{Bad Event}]$

$= \Pr\left[\begin{array}{c}\text{Bad Event}\\ \text{in Phase 1}\end{array}\right] + \Pr\left[\begin{array}{c}\text{Bad Event}\\ \text{in Phase 2}\end{array}\right]$

$\leq \Pr[\text{Bad Event in Phase 1}] + \Pr[\text{Bad Event in Phase 2}]$

$\leq \frac{1}{n^{c-2}} + \frac{1}{n^{100}}$

$\leq \frac{1}{n^{98}}$ (for $c = 101$)