Sampling Edges almost u.a.r.

Design an alg s.t. each edge is returned with almost equal prob: $\frac{1 \pm \varepsilon}{m}$.

Refer to such a distribution as pointwise ε-close to uniform.

Motivation:

Sampling edges is a very basic primitive appearing in many sublinear-time algorithms.

For example, several subgraph estimation algs use it as a basic query.

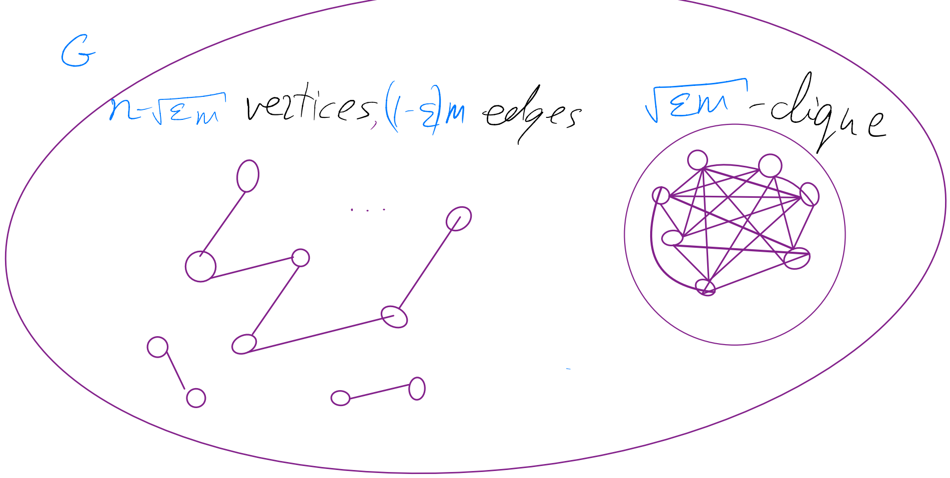
Why pointwise-close and not, say, ℓ_1-close?

Two distributions P, Q are

Pointwise ε-close if $\forall x \in \Omega,\ |P(x) - Q(x)| \leq \varepsilon P(x)$

ℓ_1 ε-close if $\sum_{x \in \Omega} |P(x) - Q(x)| \leq \varepsilon |\Omega|$

In particular, ℓ_1-closeness allows to set $P(x) = 0$ for an ε-fraction of the elements in the domain



Say we want estimate the number of triangles, T using close-to-uniform edge samples

Given access to only ℓ_1 ε-close to uniform samples, we might never see triangles

(Using pointwise ε-close samples, this could be done in $\tilde{O}\left(\frac{m^{3/2}}{T}\right)$ expected queries)

Query Model: Adjacency list :

Query Model: Adjacency list:

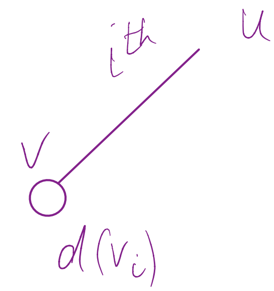
- The vertices are labeled arbitrarily $1..n$, and Alg knows n.

Query Model: Adjacency list:

- The vertices are labeled arbitrarily $1..n$, and Alg knows n.
- Degree queries: $deg(v)$ returns d_v.

Query Model: Adjacency list:

- The vertices are labeled arbitrarily $1..n$, and Alg knows n.



- Degree queries: $deg(v)$ returns d_v.
- Neighbor queries: $nbr(v, i)$ returns the i^{th} neighbor of v, if one exists. O.w. returns $\perp$.

Easy case:

Consider a d-regular graph G

Easy case:

Consider a d-regular graph G

Think of each edge $\{u,v\}$ as two oriented edges $\overrightarrow{(u,v)}$ & $\overrightarrow{(v,u)}$

Easy case:

Consider a d-regular graph G

Think of each edge $\{u,v\}$ as two oriented edges $\overrightarrow{(u,v)}$ & $\overrightarrow{(v,u)}$

To sample an oriented edge u.a.r. do:

1. Sample $u \in V$ u.a.r.

Easy case:

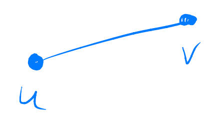
Consider a d-regular graph G

Think of each edge $\{u,v\}$ as two oriented edges $\overrightarrow{(u,v)}$ & $\overrightarrow{(v,u)}$

To sample an oriented edge u.a.r. do:

1. Sample $u \in V$ u.a.r.

2. Sample a nbr v of u u.a.r



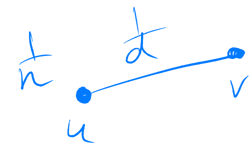
Easy case:

Consider a d-regular graph G

Think of each edge $\{u,v\}$ as two oriented edges $\overrightarrow{(u,v)}$ & $\overrightarrow{(v,u)}$

To sample an oriented edge u.a.r. do:

$$\boxed{\begin{array}{l} \text{1. Sample } u \in V \text{ u.a.r.} \\ \text{2. Sample a nbr } v \text{ of } u \text{ u.a.r} \end{array}}$$



For every edge $\overrightarrow{(u,v)}$ $\Pr\left[\overrightarrow{(u,v)}\right] = \frac{1}{n} \cdot \frac{1}{d} = \frac{1}{m}$

Now assume G is a graph with max deg. d_{max}.

Now assume G is a graph with max deg. d_{max}.

We can try to sample an edge u.a.r. as follows.

Now assume G is a graph with max deg. d_{max}.

We can try to sample an edge u.a.r. as follows.

1. Sample $u \in V$ u.a.r. 2. Choose $i \in [d_{max}]$ u.a.r. 3. Query for the i^{th} nieghbor of u. 4. If an edge was returned, then return it. O.w. fail.

Now assume G is a graph with max deg. d_{max}.

We can try to sample an edge u.a.r. as follows.

1. Sample $u \in V$ u.a.r.
2. Choose $i \in [d_{max}]$ u.a.r.
3. Query for the i^{th} nieghbor of u.
4. If an edge was returned, then return it. O.w. fail.

For every $\overrightarrow{(u,v)}$: $\Pr[\overrightarrow{(u,v)} \text{ returned}] = \frac{1}{n} \cdot \frac{1}{d_{max}}$

Prob of u

Prob. that i is the index of v

Now assume G is a graph with max deg. d_{max}.

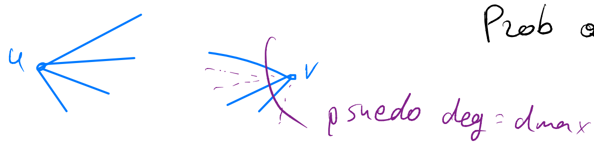
We can try to sample an edge u.a.r. as follows.

1. Sample $u \in V$ u.a.r.
2. Choose $i \in [d_{max}]$ u.a.r.
3. Query for the i^{th} nieghbor of u.
4. If an edge was returned, then return it. O.w. fail.

For every $\overrightarrow{(u,v)}$: $Pr[\overrightarrow{(u,v)} \text{ returned}] = \frac{1}{n} \cdot \frac{1}{d_{max}}$

Prob of u

Prob that i is the index of v



Observe that:

1. Each edge is sampled with equal prob. $\frac{1}{n \cdot d_{max}}$

2. Success prob. of a single invocation is

$$\sum_{(u,v) \in E} \Pr[\overrightarrow{(u,v)} \text{ returned}] = \frac{2m}{n d_{max}}$$

$\Rightarrow$ # required attempts $= O\left(\frac{n d_{max}}{m}\right)$

If $n \cdot d_{max}$ is much higher than m, then this alg. is not efficient. (could be a high as $O(\frac{n^2}{m})$)

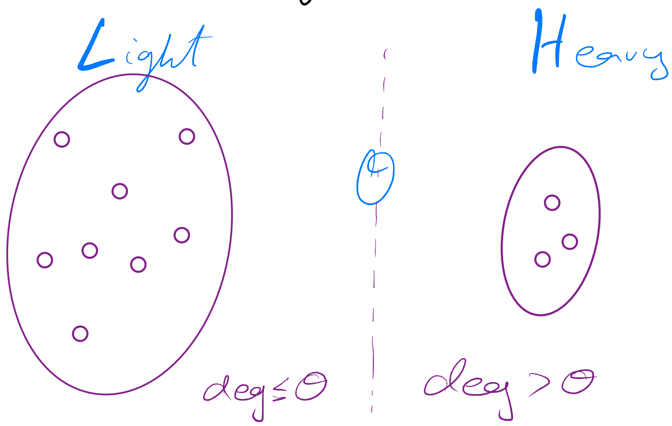
1. Think of every edge $\{u,v\}$ as two oriented edges (u,v)
$+ (v,u)$

1. Think of every edge $\{u,v\}$ as two oriented edges (u,v)
& (v,u)

2. Partition the graph vertices into **Light** & **Heavy** vertices according to a deg. threshold θ

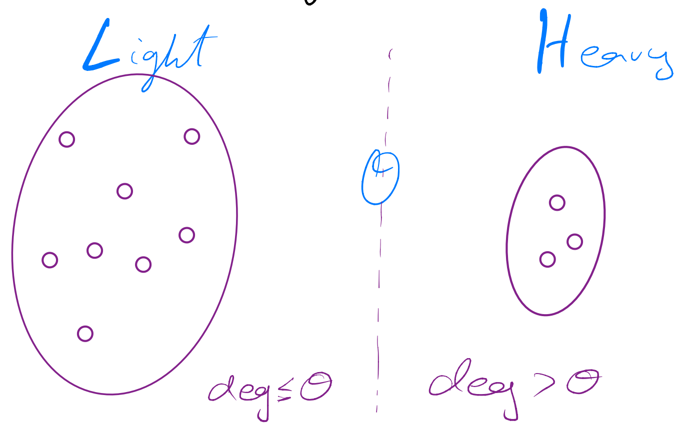
1. Think of every edge $\{u,v\}$ as two oriented edges (u,v) & (v,u)

2. Partition the graph vertices into **Light** & **Heavy** vertices according to a deg. threshold $\theta = \left(\sqrt{2m/\varepsilon}\right)$



1. Think of every edge $\{u,v\}$ as two oriented edges (u,v) & (v,u)

2. Partition the graph vertices into **Light** & **Heavy** vertices according to a deg. threshold $\theta = \left(\sqrt{2m/\varepsilon}\right)$



3. For every v, let $\Gamma_L(v) = \Gamma(v) \cap \mathbf{L}$ & $\Gamma_H(v) = \Gamma(v) \cap \mathbf{H}$

Try-to-Sample-Edge

Try-to-Sample-Ed e

w.p. $\boxed{?}$

(Try-to-) Sample-Light(θ)

Try-to-Sample-Edge

w.p. $\boxed{?}$ (Try-to-) Sample-Light(θ)

w.p. $\boxed{?}$ (Try-to-) Sample-Heavy(θ)

Try-to-Sample-Edge

w.p. $\boxed{?}$ w.p. $\boxed{?}$

(Try-to-) Sample-**L**ight(θ) (Try-to-) Sample-**H**eavy(θ)

1. Sample $u \in V$ u.a.r

Try-to-Sample-Edge

w.p. $\boxed{?}$ w.p. $\boxed{?}$

(Try-to-) Sample-Light(θ) (Try-to-) Sample-Heavy(θ)

1. Sample $u \in V$ u.a.r
2. Query $d(u)$

Try-to-Sample-Edge

w.p. $\boxed{?}$ w.p. $\boxed{?}$

(Try-to-) Sample-**L**ight(θ) (Try-to-) Sample-**H**eavy(θ)

1. Sample $u \in V$ u.a.r
2. Query $d(u)$
3. If $d(u) > \theta$, fail

Try-to-Sample-Edge

w.p. $\boxed{?}$ ↙ ↘ w.p. $\boxed{?}$

(Try-to-) Sample-**Light**(θ) (Try-to-) Sample-Heavy(θ)

1. Sample $u \in V$ u.a.r
2. Query $d(u)$
3. If $d(u) > \theta$, fail
4. Choose $i \in [\theta]$ uar.

Try-to-Sample-Edge

w.p. $\boxed{?}$ w.p. $\boxed{?}$

(Try-to-) Sample-Light(θ) (Try-to-) Sample-Heavy(θ)

1. Sample $u \in V$ u.a.r
2. Query $d(u)$
3. If $d(u) > \theta$, fail
4. Choose $i \in [\theta]$ uar.
5. Query for the i^{th} nbr of u. Let v denote the returned nbr if one was returend. O.w. fail.

Try-to-Sample-Edge

w.p. $\boxed{?}$ w.p. $\boxed{?}$

(Try-to-) Sample-Light(θ) (Try-to-) Sample-Heavy(θ)

1. Sample $u \in V$ u.a.r
2. Query $d(u)$
3. If $d(u) > \theta$, fail
4. Choose $i \in [\theta]$ uar.
5. Query for the i^{th} nbr of u. Let v denote the returned nbr if one was returend. O.w. fail.
6. Return $\overrightarrow{(u,v)}$

Analysis of Sample-Light:

Fix a light edge $\overrightarrow{(u,v)}$

$\Pr[(u,v) \text{ is returned}]$

$= \Pr[u \text{ is chosen in } 1] \cdot \Pr[\text{the index of } v \text{ is chosen in } 4.]$

$= \frac{1}{n} \cdot \frac{1}{\theta} = \frac{1}{n\theta}$

Sample-Light(θ)

1. Sample $u \in V$ u.a.r
2. Query $d(u)$
3. If $d(u) > \theta$, fail
4. Choose $i \in [\theta]$ uar.
5. Query for the i^{th} nbr of u. Let v denote the returned nbr if one was returned. O.w. fail.
6. Return $\overrightarrow{(u,v)}$

Try-to-Sample-Edge

w.p. $\boxed{?}$

(Try-to-) Sample-Light(θ)

1. Sample $u \in V$ u.a.r
2. Query $d(u)$
3. If $d(u) > \theta$, fail
4. Choose $i \in [\theta]$ uar.
5. Query for the i^{th} nbr of u. Let v denote the returned nbr if one was returend. O.w. fail.
6. Return $\overrightarrow{(u,v)}$

w.p. $\boxed{?}$

(Try-to-) Sample-Heavy (θ)

1. Invoke Sample-Light

Try-to-Sample-Edge

w.p. $\boxed{?}$ w.p. $\boxed{?}$

(Try-to-) Sample-**L**ight(θ)

1. Sample $u \in V$ u.a.r
2. Query $d(u)$
3. If $d(u) > \theta$, fail
4. Choose $i \in [\theta]$ u.a.r.
5. Query for the i^{th} nbr of u. Let v denote the returned nbr if one was returend. O.w. fail.
6. Return $\overrightarrow{(u,v)}$

(Try-to-) Sample-**H**eavy(θ)

1. Invoke Sample-Light
2. Let $\overrightarrow{(u,v)}$ be the returned edge

Try-to-Sample-Edge

w.p. $\boxed{?}$ w.p. $\boxed{?}$

(Try-to-) Sample-Light(θ)

1. Sample $u \in V$ u.a.r
2. Query $d(u)$
3. If $d(u) > \theta$, fail
4. Choose $i \in [\theta]$ u.a.r.
5. Query for the i^{th} nbr of u. Let v denote the returned nbr if one was returend. O.w. fail.
6. Return $\overrightarrow{(u,v)}$

(Try-to-) Sample-Heavy(θ)

1. Invoke Sample-Light
2. Let $\overrightarrow{(u,v)}$ be the returned edge
3. If $d(v) \leq \theta$, fail

Try-to-Sample-Edge

w.p. ? w.p. ?

(Try-to-) Sample-Light(θ)

1. Sample $u \in V$ u.a.r
2. Query $d(u)$
3. If $d(u) > \theta$, fail
4. Choose $i \in [\theta]$ u.a.r.
5. Query for the i^{th} nbr of u. Let v denote the returned nbr if one was returned. O.w. fail.
6. Return $\overrightarrow{(u,v)}$

(Try-to-) Sample-Heavy(θ)

1. Invoke Sample-Light
2. Let $\overrightarrow{(u,v)}$ be the returned edge
3. If $d(v) \le \theta$, fail
4. Query a uniform nbr of v, w

Try-to-Sample-Edge

w.p. $\boxed{?}$ ↙ w.p. $\boxed{?}$ ↘

(Try-to-) Sample-Light(θ)

1. Sample $u \in V$ u.a.r
2. Query $d(u)$
3. If $d(u) > \theta$, fail
4. Choose $i \in [\theta]$ u.a.r.
5. Query for the i^{th} nbr of u. Let v denote the returned nbr if one was returend. O.w. fail.
6. Return $\overrightarrow{(u,v)}$

(Try-to-) Sample-Heavy (θ)

1. Invoke Sample-Light
2. Let $\overrightarrow{(u,v)}$ be the returned edge
3. If $d(v) \leq \theta$, fail
4. Query a uniform nbr of v, w
5. Return $\overrightarrow{(v,w)}$

Try-to-Sample-Edge

w.p. $\boxed{?}$

(Try-to-) Sample-Heavy (θ)

1. Invoke Sample-Light
2. Let $\overrightarrow{(u,v)}$ be the returned edge
3. If $d(v) \leq \theta$, fail
4. Query a uniform nbr of v, w
5. Return $\overrightarrow{(v,w)}$

Intuition:

We will try to "reach" the high deg. vertices from their low deg. neighbors:

* Vertices in H have high deg $(> \sqrt{2m/\varepsilon})$ and we'll show that most of their edges are coming from low degree nbrs.

* We already know how to sample edges originating in low deg vertices

Analysis of Sample-Heavy:

Sample-Heavy (θ)

1. Invoke Sample-Light
2. Let $\overrightarrow{(u,v)}$ be the returned
3. If $d(v) \leq \theta$, fail
4. Query a uniform nbr of v
5. Return $\overrightarrow{(v,w)}$

Fix heavy edge $\overrightarrow{(v,w)}$

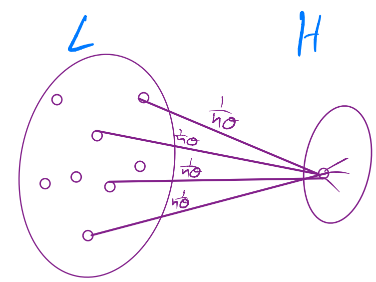
$\Pr[\overrightarrow{(v,w)}$ is returned$] =$

$\Pr[$an edge (u,v) is returned in 1.$] \cdot \Pr[w$ is sampled in 4.$]$

$$= \frac{|\Gamma_L(v)|}{n\theta} \cdot \frac{1}{d(v)}$$

since v has $|\Gamma_L(v)|$ incoming light edges,

and by previous analysis, each is returned by Sample-Light w.p. $\frac{1}{n\theta}$



Analysis of Sample-Heavy:

Sample-Heavy(θ)

1. Invoke Sample-Light
2. Let $\overrightarrow{(u,v)}$ be the returned
3. If $d(v) \leq \theta$, fail
4. Query a uniform nbr of v
5. Return $\overrightarrow{(v,w)}$

Fix heavy edge $\overrightarrow{(v,w)}$

$\Pr[\overrightarrow{(v,w)}$ is returned$]$ =

$\Pr[$an edge (u,v) is returned in 1.$] \cdot \Pr[w$ is sampled in 4.$]$

$$= \frac{|\Gamma_L(v)|}{n\theta} \cdot \frac{1}{d(v)}$$

Proof follows.

$$\geq \frac{(1-\varepsilon)\,d(v)}{n\theta} \cdot \frac{1}{d(v)} = \frac{1-\varepsilon}{n\theta}$$

Sample-Heavy(θ)
1. Invoke Sample-Light
2. Let $(\overrightarrow{u,v})$ be the returned
3. If $d(v) \le \theta$, fail
4. Query a uniform nbr of v
5. Return $(\overrightarrow{v,w})$

Analysis of Sample-Heavy:

Fix heavy edge $(\overrightarrow{v,w})$

$\Pr[(\overrightarrow{v,w})$ is returned$] =$

$\Pr[$an edge (u,v) is returned in 1.$] \cdot \Pr[w$ is sampled in 4.$]$

$$= \frac{|\Gamma_L(v)|}{n\theta} \cdot \frac{1}{d(v)}$$

$$\geq \frac{(1-\varepsilon)\, d(v)}{n\theta} \cdot \frac{1}{d(v)} = \frac{1-\varepsilon}{n\theta}$$

$$\leq \frac{d(v)}{n\theta} \cdot \frac{1}{d(v)} = \frac{1}{n\theta}$$

Sample-Heavy(θ)

1. Invoke Sample-Light
2. Let $(\overrightarrow{u,v})$ be the returned
3. If $d(v) \le \theta$, fail
4. Query a uniform nbr of v
5. Return $(\overrightarrow{v,w})$

Analysis of Sample-Heavy:

Fix heavy edge $(\overrightarrow{v,w})$

$\Pr[(\overrightarrow{v,w}) \text{ is returned}] =$

$\Pr[\text{an edge } (u,v) \text{ is returned in } 1.] \cdot \Pr[w \text{ is sampled in } 4.]$

$$= \frac{|\Gamma_L(v)|}{n\theta} \cdot \frac{1}{d(v)}$$

$$\geq \frac{(1-\varepsilon)\, d(v)}{n\theta} \cdot \frac{1}{d(v)} = \frac{1-\varepsilon}{n\theta}$$

$$\leq \frac{d(v)}{n\theta} \cdot \frac{1}{d(v)} = \frac{1}{n\theta}$$

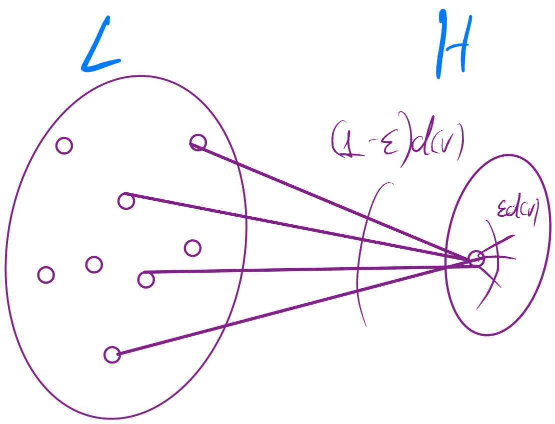
$$\in \left[\frac{1-\varepsilon}{n\theta}, \frac{1}{n\theta}\right]$$

The setting of θ

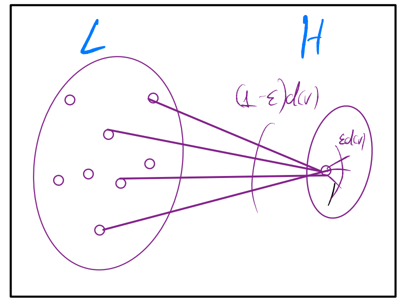
The setting of θ

Claim: by setting $\theta = \sqrt{2m/\varepsilon}$, we get

$d_L(v) = |\Gamma_L(v)| \geq (1-\varepsilon)d(v)$.



The setting of θ



Claim: by setting $\theta = \sqrt{2m/\varepsilon}$, we get

$$d_L(v) = |\Gamma_L(v)| \geq (1-\varepsilon)\, d(v).$$

Proof: It holds that $|H| \leq \frac{2m}{\theta} = \sqrt{\varepsilon 2m}$

Also, $\forall\, w \in \Gamma(v)$, $w \in H$.

Hence, $d_H(v) = |\Gamma_H(v)| \leq |H| \leq \sqrt{\varepsilon 2m} = \varepsilon\theta \leq \varepsilon\, d(v)$

$\Rightarrow \quad d_L(v) = d(v) - d_H(v) \geq (1-\varepsilon)\, d(v)$

$\blacksquare$

Try-to-Sample-Edge

w.p. $\boxed{?}$ w.p. $\boxed{?}$

(Try-to-) Sample-Light(θ)

1. Sample $u \in V$ u.a.r
2. Query $d(u)$
3. If $d(u) > \theta$, fail
4. Choose $i \in [\theta]$ u.a.r.
5. Query for the i^{th} nbr of u. Let v denote the returned nbr if one was returned. O.w. fail.
6. Return $\overrightarrow{(u,v)}$

$\frac{1}{n\theta}$

(Try-to-) Sample-Heavy (θ)

1. Invoke Sample-Light
2. Let $\overrightarrow{(u,v)}$ be the returned edge
3. If $d(v) \leq \theta$, fail
4. Query a uniform nbr of v, w
5. Return $\overrightarrow{(v,w)}$

$\in \left[\frac{1-\varepsilon}{n\theta}, \frac{1}{n\theta}\right]$

Try-to-Sample-Edge

w.p. $\frac{1}{2}$ → (Try-to-) Sample-Light(θ)

w.p. $\frac{1}{2}$ → (Try-to-) Sample-Heavy(θ)

(Try-to-) Sample-Light(θ)

1. Sample $u \in V$ u.a.r
2. Query $d(u)$
3. If $d(u) > \theta$, fail
4. Choose $i \in [\theta]$ uar.
5. Query for the i^{th} nbr of u. Let v denote the returned nbr if one was returend. O.w. fail.
6. Return $\underbrace{\overrightarrow{(u,v)}}_{\frac{1}{n\theta}}$

(Try-to-) Sample-Heavy(θ)

1. Invoke Sample-Light
2. Let $\overrightarrow{(u,v)}$ be the returned edge
3. If $d(v) \leq \theta$, fail
4. Query a uniform nbr of v, w
5. Return $\underbrace{\overrightarrow{(v,w)}}_{\in \left[\frac{1-\varepsilon}{n\theta}, \frac{1}{n\theta}\right]}$

Hence, Try-to-Sample-Edge returns each oriented edge in the graph w.p.

in $\frac{1}{2} \cdot \left[\frac{1-\varepsilon}{n\theta}, \frac{1}{n\theta} \right]$ for $\theta = \sqrt{2m/\varepsilon}$

Hence, Try-to-Sample-Edge returns each oriented edge in the graph w.p.

in $\frac{1}{2} \cdot \left[\frac{1-\varepsilon}{n\theta}, \frac{1}{n\theta} \right]$ for $\theta = \sqrt{2m/\varepsilon}$

Therefor, the overall success prob. of a single invocation is $\sum_{(u,v) \in \vec{E}} \Pr[(\vec{u,v}) \text{ returned}] \in \left[\frac{(1-\varepsilon)m}{2n\theta}, \frac{m}{2n\theta} \right]$.

Hence, Try-to-Sample-Edge returns each oriented edge in the graph w.p. in $\frac{1}{2} \cdot \left[\frac{1-\varepsilon}{n\theta}, \frac{1}{n\theta} \right]$ for $\theta = \sqrt{2m/\varepsilon}$.

Therefor, the overall success prob. of a single invocation is $\sum_{(\vec{u,v}) \in \vec{E}} \Pr\left[(\vec{u,v}) \text{ returned} \right] \in \left[\frac{(1-\varepsilon)m}{2n\theta}, \frac{m}{2n\theta} \right]$.

Hence, expected # of sufficient invocations until an edge is returned is $\frac{2n\theta}{m} = O\left(\frac{n}{\sqrt{\varepsilon m}}\right)$

(where before we had $O\left(\frac{n \cdot d_{max}}{m}\right) = O\left(\frac{n^2}{m}\right)$)

Sample-an-Edge (m, ε)

1. Set $\theta = \sqrt{2m/\varepsilon}$
2. Repeatedly invoke Try-to-Sample-Edge(θ) until an edge $(\vec{x, y})$ is returned
3. Return $(\vec{x, y})$

Sample-an-Edge (m, ε)

1. Set $\theta = \sqrt{2m/\varepsilon}$
2. Repeatedly invoke Try-to-Sample-Edge(θ) until an edge $(\vec{x, y})$ is returned
3. Return $(\vec{x, y})$

By the above:

1. Each edge is returned w.p. $\frac{1 \pm \varepsilon}{m}$

Sample-an-Edge (m, ε)

1. Set $\theta = \sqrt{2m/\varepsilon}$
2. Repeatedly invoke Try-to-Sample-Edge(θ) until an edge $(\vec{x, y})$ is returned
3. Return $(\vec{x, y})$

By the above:

1. Each edge is returned w.p. $\frac{1 \pm \varepsilon}{m}$
2. The expected query and time complexities of Sample-an-Edge are $O\left(\frac{n}{\sqrt{\varepsilon m}}\right)$.