Note: Tim no longer thinks this works. Each edge exists with probability

$$\frac{w_e r_e(H_t)}{blah}$$

times some $\log n\varepsilon^{-2}$ factor. Howver, each edge is not an independent coin-flip. We attempt to recreate semi-streaming sparsification (knowing now that it's possible) without the general framework of resparsification.

Note on Spielman (regular): (No weights) Using per-edge Spielman Srivastava, we know we want $n \log n\varepsilon^{-2}$ edges in the end, so we keep edges with probability $\frac{r_e \log n}{\varepsilon^2}$. (Reweighting is the reciprocal of this probability.)

Therefore in resparsification, we would (ideally) like to not lose edges that are chosen with probabilty 1, which is equivalent to not losing edges with

$$\frac{r_e \log n}{\varepsilon^2} > 1 \tag{1}$$

$$\tag{2}$$

When the first $2T$ edges of the semi-stream is graph H, then our algorithm: flips coins on edges with $\frac{r_e(H)\log n}{\varepsilon^2} < 1$ with probability equal to that number. Otherwise it keeps the edge in by default.

Here, $T = \frac{n \log n}{\varepsilon^2}$.

Lemma 0.1. *No edges with* $\frac{r_e(G)\log n}{\varepsilon^2} > 1$ *are perturbed by the above algorithm*

Perturbed edges in the algorithm are edges where $\frac{r_e(H)\log n}{\varepsilon^2} < 1$. We show that any edge with this property also satisfies the same property for G.

Proof. Note that all edges e with

$$\frac{r_e(H)\log n}{\varepsilon^2} < 1$$

also satisfy:

$$\begin{aligned}&\frac{r_e(G)\log n}{\varepsilon^2}\\ &\leq \frac{r_e(H)\log n}{\varepsilon^2}\\ &< 1\end{aligned}$$

because for any subset $H \subset G$,

$r_e(G) \leq r_e(H)$.

Therefore, no perturbed edges has $\frac{r_e(G)\log n}{\varepsilon^2} > 1$.

□

1 Semi-Streaming sparsification without Predictable Quadratic Variance

Now we prove that semi-streaming sparisification as described in the resparsification game paper of KPPS works, but without using predictable quadratic variation. All we really do is show that the algorithm in $KPPS$ zeros out an edge with low probability, enough to satisfy the guarantees of Spielman Srivastava sparsification.

To do this, we will use two lemmas and an induction argument.

Lemma 1.1. *For $H \subset G$, we have $r_e(H) \leq r_e(G)$.*

Lemma 1.2. *(Known variation on Spielman Srivastava). For any graph G, suppose $\boldsymbol{r}_e$ is a 2-approximation on the effective resistance r_e in G. Let w_e denote the weight of edge e. For each edge e, flip a coin with probability :*

$$C \cdot \frac{w_e \boldsymbol{r}_e \log n}{\varepsilon^2} \tag{3}$$

and with that probability weight the edge with weight

$$\frac{\varepsilon^2}{\boldsymbol{r}_e \log n} \tag{4}$$

then the resulting graph is a $1+\varepsilon$-sparsifier of G with high probability, for some constant C. The number of edges in this graph is roughly $Cn \log n \varepsilon^{-2}$ edges.

We outline the semi-streaming algorithm used in this note, which is essentially identical to the one in KPPS. Let T be $2C \cdot \left(\frac{n \log n}{\varepsilon^2}\right)$. Let G_k be the subgraph formed by the first k edges of our stream. Let the k^{th} edge in the stream be E_k. (Here, $G_m =: G$, where G is the graph formed by all edges of the stream).

The algorithm is:

This gets you under T edges with high probability. We will prove a few lemmas about this algorithm inductively, which will show its a good sparsifier with high probability.

Lemma 1.3. *H_{k-1} as defined in the algorithm is a good $1+\varepsilon$ sparsifier of G_k for all k.*

We will prove this lemma by induction. For this, we need:

Lemma 1.4. *The probability edge $e := E_{k'}$ is non-zero at time $k > k'$ is bounded below by:*

$$(1-\varepsilon)\frac{w_e r_e(H_k) \log n}{\varepsilon^2} \tag{5}$$

Algorithm 1 SEMISTREAMSPARSIFY(G, ε)

Input: Positively weighted graph G with m edges adn n vertices.

1. While $k < m$:
   (a) $H_k \leftarrow H_{k-1} + E_k$.
   (b) If $|E(H_k)| > 2T$:
      i. Flip a coin on each edge with probability $C \cdot \frac{w_e r_e(H_k) \log n}{\varepsilon^2}$ and weight it with: $\frac{\varepsilon^2}{r_e(H_k)}$.
   (c) H_k is the output of the above procedure.

IF $H_k = H_{k-1} + E_k$, then Lemma **??** follows from H_{k-1} being a good sparsifier of G_{k-1}.

If H_k undergoes a re-sparsification step (step **??** in SEMISTREAMSPARSIFY then the probability any edge $e := E_{k'}$ is non-zero in the final value of H_k, for $k' \leq k$, is equal to

$$\prod_{t \in T_{k',k}} \left(w_{e,t} \cdot r_e(H_t) \cdot \frac{\log n}{\varepsilon^2} \right) \tag{6}$$

where $T_{k',k}$ represents the set of times $t \leq k$ where $E_{k'}$ was sparsified. Here, $w_{e,t}$ is the weight of edge t in H_t, and $r_e(H_t)$ is the effective resistance of H_t.

However, note that $w_{e,t} = w_e$ if t is the first time e is resparsified, and $\frac{\varepsilon^2}{r_e(H_{t-1})}$ otherwise. Therefore, the whole product in Equation (**??**) telescopes into:

$$w_e r_e(H_{t'}) \frac{\log n}{\varepsilon^2} \tag{7}$$

where t' is the last time $E_{k'}$ was sparsified before time k. If $H_{t'}$ is a spectral sparsifier of $G_{t'}$, then we're done. However, this is true because ;q

Oops this doens't work because all I did was bound the probability an edge was picked, but I did not successfully show the edge picks are independent! (And indeed they are not!)