Lecture Notes CS:5350 Design and Analysis of Algorithms
Lecture 13: Oct 8, 2019
Scribe: Hankyu Jang

1 Luby's Algorithm for Maximal Independent Set (MIS)

In the previous lecture, we proved that high degree nodes get deactivated with at least constant probability in the first iteration of Luby's MIS algorithm. This is stated in the following lemma:

Lemma 1 *Let Δ be the maximum degree of the graph. Let v be a vertex with degree $\geq \frac{\Delta}{2}$. Then, v is deactivated in iteration 1 with probability $\geq \left(1 - \frac{1}{e^{\frac{1}{4}}}\right) \cdot \frac{1}{2}$*

In today's lecture, using this result, we will prove that Luby's MIS algorithm terminates in $O(\log n \log \Delta)$ rounds, with high probability, where n denotes the number of nodes in the graph.

1.1 Analysis for a particular vertex v

We now strengthen Lemma 1 to apply to all iterations. Note that the notation Δ is being used in slightly different ways in Lemma 1 and Lemma 2.

Lemma 2 *Let Δ be an upper bound on the maximum degree of the graph induced by active nodes just before iteration i. Let us call this the **active graph**. Let v be a vertex with degree $\geq \frac{\Delta}{2}$ in this active graph. Then, v is deactivated in iteration i with probability $\geq \left(1 - \frac{1}{e^{\frac{1}{4}}}\right) \cdot \frac{1}{2}$*

A key point in Lemma 2 is that, in any iteration, the probability of a high degree vertex being deactivated is at least a constant.
Note: The original graph in Lemma 1 or the active graph in Lemma 2 need not be connected. In other words, there may be completely isolated nodes in the graph.

Let us apply Lemma 2 to the rest of the proof to show that the algorithm terminates in $O(\log n \log \Delta)$ iterations with high probability.

Lemma 3 *Let Δ be the maximum degree of the original graph. Let v be a vertex with degree $\geq \frac{\Delta}{2}$. Then, for some large enough constant C, v is either deactivated or v's degree in the active becomes $< \frac{\Delta}{2}$ in $C \log n$ iterations with probability $\geq 1 - \frac{1}{n^3}$.*

An essential aspect in Lemma 3 is that repeating an iteration $C \log n$ times for a high degree node v amplifies the probability of that node being deactivated. In other words, this repetition brings down the survival probability for high degree nodes. Amplifying the probability of an event by simply repeating the iterations is a critical component of randomized algorithms.

Take a look at the following example to understand this clearly. Assume that the probability of a good event happening is $\frac{1}{2}$. How many independent iterations would it take to ensure that the probability of the good event not happening is down to $\frac{1}{n^3}$? The answer is $3 \log_2 n$ times, because $(\frac{1}{2})^{3 \log_2 n} = \frac{1}{n^3}$.

Keeping this in mind, let us prove the Lemma 3.

Proof: Without loss of generality (WLOG), let $C\log n$ be a positive integer and let T denote $C\log n$. Let E_i denote the event that v is not deactivated and v has degree $\geq \frac{\Delta}{2}$ in the active graph at the end of iteration i.

We want an upperbound on $Prob[E_1 \wedge E_2 \wedge \cdots \wedge E_T]$. Based on the *chain rule* on the conditional probability,

$$\begin{aligned} Prob[E_1 \wedge E_2 \wedge \cdots \wedge E_T] = Prob[E_1] \cdot Prob[E_2|E_1] \cdot Prob[E_3|E_1 \wedge E_2] \cdot \\ \cdots \cdot Prob[E_T|E_1 \wedge E_2 \wedge \cdots \wedge E_{T-1}] \end{aligned} \tag{1}$$

As we will be upperbounding Equation 1 , we need to fully understand each component in the R.H.S. of Equation 1. Following is the explanation of these components:

- $Prob[E_1]$: Probability of a "bad event" happening in iteration 1. Here, a "bad event" refers to having no progress with respect to v; in other words, if v continues to be a high degree node remaining active after the iteration 1, we regard that a "bad event" happened. An upper bound can be imposed based on Lemma 2:

$$Prob[E_1] \leq 1 - \left(1 - \frac{1}{e^{\frac{1}{4}}}\right) \cdot \frac{1}{2}$$

- $Prob[E_2|E_1]$: Probability of a "bad event" happening in iteration 2 given that a "bad event" happened in iteration 1. Similarly, from Lemma 2:

$$Prob[E_2|E_1] \leq 1 - \left(1 - \frac{1}{e^{\frac{1}{4}}}\right) \cdot \frac{1}{2}$$

- $Prob[E_3|E_1 \wedge E_2]$: Probability of a "bad event" happening in iteration 3 given that a "bad event" happened in iteration 1 and iteration 2. Similarly, from Lemma 2:

$$Prob[E_3|E_1 \wedge E_2] \leq 1 - \left(1 - \frac{1}{e^{\frac{1}{4}}}\right) \cdot \frac{1}{2}$$

- $Prob[E_T|E_1 \wedge E_2 \wedge \cdots \wedge E_{T-1}]$: Probability of a "bad event" happening in iteration T given that a "bad event" happened in all of the previous iterations. Similarly, from Lemma 2:

$$Prob[E_T|E_1 \wedge E_2 \wedge \cdots \wedge E_{T-1}] \leq 1 - \left(1 - \frac{1}{e^{\frac{1}{4}}}\right) \cdot \frac{1}{2}$$

The R.H.S. of Equation 1 is the multiplication of probabilities and the upper bound for each probability is $1 - \left(1 - \frac{1}{e^{\frac{1}{4}}}\right) \cdot \frac{1}{2}$. Therefore, following inequality holds for $Prob[E_1 \wedge E_2 \wedge \cdots \wedge E_T]$:

$$Prob[E_1 \wedge E_2 \wedge \cdots \wedge E_T] \leq \left(1 - \left(1 - \frac{1}{e^{\frac{1}{4}}}\right) \cdot \frac{1}{2}\right)^T \tag{2}$$

Using the fact that $(\frac{1}{c})^{C\log n} \leq \frac{1}{n^3}$ for $c > 1$ and for sufficiently large C, we can get an upper bound on Equation 2 as:

$$Prob[E_1 \wedge E_2 \wedge \cdots \wedge E_T] \leq \frac{1}{n^3} \tag{3}$$

□

1.2 Analysis for all high degree vertices

The previous proof was focused on a particular vertex v. Now let us look at all vertices with high degree.

Lemma 4 *Let Δ be the maximum degree of the initial graph. Then, for a sufficiently large constant C, there exists a vertex of degree $\geq \frac{\Delta}{2}$ in the active graph after $C \log n$ iterations with probability at most $\frac{1}{n^2}$.*

Let $v_1, v_2, \cdots, v_k$ denote high degree nodes. We are interested in the probability that either v_1 or v_2 or $\cdots$ or v_k survives after $C \log n$ iterations. The upper bound on the survival probability of each of these nodes is $\frac{1}{n^3}$ by Lemma 3. By union bound, we can bound this probability by $\frac{1}{n^2}$. Let us now formally prove Lemma 4.

Proof: Let $v_1, v_2, \cdots, v_k$ be vertices with degree $\geq \frac{\Delta}{2}$ at the start of the algorithm. Also, let there be two events E and E_{v_i} such that

- $E \equiv$ there exists a vertex of degree $\geq \frac{\Delta}{2}$ in the active graph after $C \log n$ iterations
- $E_{v_i} \equiv v_i$ has degree $\geq \frac{\Delta}{2}$ in the active graph after $C \log n$ iterations

It is trivial that

$$E \equiv \bigvee_{i=1}^{k} E_{v_i} \tag{4}$$

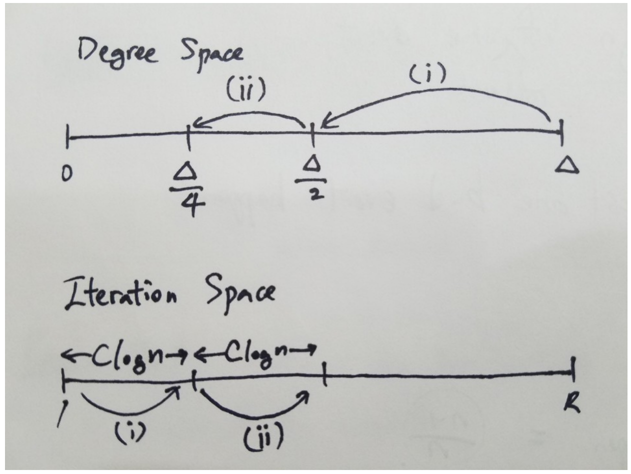
Therefore, we can bound the *Prob[E]* as follows:

$$\begin{aligned}
Prob[E] &= Prob[\bigvee_{i=1}^{k} E_{v_i}] \\
&\leq \sum_{i=1}^{k} Prob[E_{v_i}] \text{ by union bound} \\
&\leq \frac{k}{n^3} \\
&\leq \frac{1}{n^2} \text{ because } k \leq n
\end{aligned} \tag{5}$$

□

1.3 Bigger picture of the algorithm

The following figure depicts what is going on during each of the $C \log n$ batches of iterations in Luby's MIS algorithm. From the figure, we can see that for each $C \log n$ batch of iterations, the degree of the high degree nodes fall by a factor of 2 with high probability ($\geq 1 - \frac{1}{n^2}$ by Lemma 4). So, after $\log \Delta$ number of batches, Δ would eventually fall to 0 with a high probability.



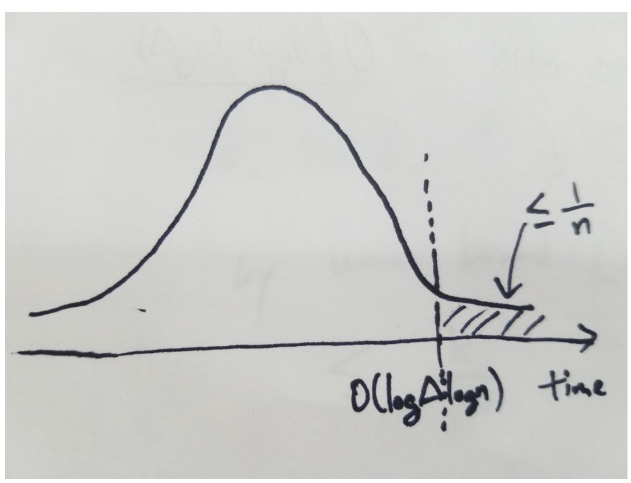
Now let us view the event that the maximum degree does not reduce by a factor of 2 in a batch of $C \log n$ iterations as a "bad event." By applying the union bound as in the proof of Lemma 4, we see that the probability that a "bad event" happens during $\log \Delta$ batches (each batch has $C \log n$ iterations) is $\frac{\log \Delta}{n^2} \leq \frac{1}{n}$. This means that the probability that no "bad event" happens is $\geq 1 - \frac{1}{n}$.

Theorem 5 *Luby's MIS algorithm computes an MIS in* $O(\log n \log \Delta)$ *rounds with probability* $\geq 1 - \frac{1}{n}$.

1.4 Comments on the algorithm

Question: Can the algorithm not compute MIS sometimes, depending on the random choices it makes?
Answer: No. It always gives a correct answer, i.e., computes an MIS. The algorithm computes MIS eventually, but with a small probability, it takes longer. In other words, Luby's MIS algorithm never produces the wrong output, but the time it takes is a random variable.

Following figure depicts a distribution of the running times of executing Luby's MIS algorithm. Almost always, Luby's MIS algorithm terminates in $O(\log \Delta \log n)$, but with a small probability $\leq \frac{1}{n}$, the algorithm may run for more iterations until the termination.



There are two types of randomized algorithms: (i) Las Vegas and (ii) Monte Carlo. Luby's algorithm is an example of Las Vegas algorithm that makes no errors, however the running time of the algorithm is a random variable. In the next class, we will start discussing these two types of randomized algorithms.