1 Overview

In the last lecture we used witness matrix to find successor matrix for APSP algorithm. Here our aim is to find a witness matrix in $O(n^w)$ time. First we look at a randomised algorithm which will run in $O(n^w \cdot \text{poly}(\log n))$ then we give a slightly different randomised algorithm which we derandomise and give a deterministic algorithm.

2 Witnessing Boolean Matrix Multiplication

We have two matrices $A$ and $B$ over $\{0, 1\}$ i.e., two boolean matrices. Our goal is to find a witness matrix $W$ such that

$$w_{ij} = \begin{cases} 
  k & \text{if } a_{ik} = 1 \text{ and } b_{kj} = 1 \\
  0 & \text{otherwise}
\end{cases}$$

2.1 Unique Witness in $O(n^w)$ time

Let $C = AB$ is the product under Boolean matrix multiplication. A witness for $c_{ij}$ is an index $k \in \{1, \ldots, n\}$ such that $a_{ik} = b_{kj} = 1$. Observe that $c_{ij} = 1$ if and only if it has some witness $k$. Let,

$$P_{ij} = \{k \mid k \text{ is a witness for } (i, j)\}$$

We want to find witnesses for all $i, j$ such that $|P_{ij}| = 1$ that is which has unique witness.

1. Compute $\hat{A}$ obtained by setting $\hat{a}_{ik} = k \cdot a_{ik}$
2. $\hat{Z} = \hat{A} \cdot B$ product over integers
3. if each entry of $C$ has a unique witness then 
   \[ W = \hat{Z} \text{ is a solution.} \]
end

Algorithm 1: Find Unique Witness

Step 3 implies that $\hat{Z}$ yields a matrix that contains witness for all entries in the matrix $C$ that have a unique witness.

Since we multiply the elements of the matrix $A$ by at most $n$, it does not affect the running time for matrix multiplication. So the running time is $O(n^w)$. 
3 Find All Witnesses

General approach of multiplying \( a_{ik} \) by \( x_k \) or by \( 2^k \) does not work in \( O(n^w) \) time because the monomeals can be long and it may take \( n^3 \) time to multiply them.

The idea is to take a subset of indices, \( S \subseteq [1, n] \) such that \( |P_{ij} \cap S| = 1 \). Then we can find the witnesses in \( S \) in \( O(n^w) \) time.

Define \( A^S \) and \( B^S \) such that each column of \( A^S \) and each row of \( B^S \) corresponding to the indices not chosen in \( S \) is turned into an all-zero vector. Suppose that the entry \( C_{ij} \) has a unique witness in the set \( S \). Then the corresponding entry in the integer matrix multiplication of \( A^S \) and \( B^S \) is the index of this unique witness.

Let, \( P_{ij} = p \). We may find the number of witnesses \( p \) by using integer matrix multiplication to compute \( Z = AB \), and then looking at the entry \( z_{ij} \). We assume that \( p \geq 2 \), since we are done, otherwise.

\[
\begin{align*}
\text{Input:} & \quad \text{Two nxn matrices } A \text{ and } B \text{ over 0,1} \\
\text{Output:} & \quad \text{Witness matrix } W \text{ for the Boolean matrix } C = AB \\
& \text{1 } W \leftarrow -AB \\
& \text{2 for } t \leftarrow 0 \text{ to log } n \text{ do} \\
& \quad r \leftarrow 2^t \\
& \quad \text{repeat} \\
& \quad \quad \text{choose random } R \subseteq 1, ..., n \text{ with } |R| = r \\
& \quad \quad \text{compute } A^R \text{ and } B^R \\
& \quad \quad Z \leftarrow A^R B^R \\
& \quad \quad \text{forall } (i, j) \text{ do} \\
& \quad \quad \quad \text{if } W_{ij} < 0 \text{ and } Z_{ij} \text{ is witness} \text{ then} \\
& \quad \quad \quad \quad W_{ij} \leftarrow Z_{ij} \\
& \quad \quad \text{end} \\
& \quad \text{end} \\
& \quad \text{until } \lceil 3.77 \log n \rceil \text{ times ;} \\
& \text{end} \\
& \text{3 forall } (i, j) \text{ do} \\
& \quad \text{if } W_{ij} < 0 \text{ then} \\
& \quad \quad \text{find witness } W_{ij} \text{ by brute force} \\
& \quad \text{end} \\
& \text{end}
\end{align*}
\]

Algorithm 2: Randomised Algorithm for BPWM

Let \( x \) be an integer such that \( n/2 \leq px \leq n \). We claim that a random set of indices \( S \subseteq \{1, ..., n\} \) is very likely to contain a unique witness i.e, \( |P_{ij}| = 1 \).

To verify this claim, consider an urn containing \( n \) balls, one for each of the \( n \) indices; the balls corresponding to witnesses are colored white, and the rest are colored black. The following lemma then shows that the probability that \( S \) contains a unique witness is reasonably large.

**Lemma 1.** Suppose an urn contains \( n \) balls of which \( w \) are white, and \( n - w \) are black. Consider choosing \( x \) balls at random (without replacement), where \( n/2 \leq px \leq n \). Then \( Pr[\text{exactly one white ball is chosen}] \geq 1/2e \).
\textbf{Proof.} Among the $x$ balls, exactly 1 can be white in $\binom{x}{w}(n-x-1)$ ways, and $x$ balls can be chosen in $\binom{x}{w}$ ways.

\[ \Pr[\text{exactly one white ball is chosen}] = \frac{\binom{x}{w}(n-x)!(n-w)!}{x!(x-1)!n!(n-w-x+1)!} \]

\[ = w.x \prod_{i=0}^{w-1} \frac{1}{(n-i)} \prod_{j=0}^{w-2} (n-x-j) \]

\[ = \frac{wx}{n} \prod_{j=0}^{w-2} \frac{n-x-j}{n-1-j} \]

\[ \geq \frac{wx}{n} \prod_{j=0}^{w-2} \frac{n-x-j-(w-j-1)}{n-1-j-(w-j-1)} \]

\[ = \frac{wx}{n} \prod_{j=0}^{w-2} \frac{n-w-(x-1)}{n-w} \]

\[ = \frac{wx}{n} (1 - \frac{n-x-j}{n-1-j})^{w-1} \]

\[ \geq 1/2(1 - \frac{1}{w})^{w-1} \geq \frac{1}{2e} \]

Therefore probability that it finds a unique witness $\geq 1/2e$

Therefore probability of failure (in $a\log n$ steps) = $(1 - 1/2e)^{a\log n}$

$= [(1 - 1/2e)^2]^{a\log n} \leq 1/n^3$

Therefore probability that the above algorithm fails for all the pair (By union bound) = $n^2 \cdot \frac{1}{n^3} = 1/n$

\subsection{Running Time:}

Let, random variable $X_{ij} = \begin{cases} 1 & \text{if we did not find an witness} \\ 0 & \text{otherwise} \end{cases}$

\[ \Pr[X_{ij} = 1] \leq 1/n^3 \]

Expected number of indeces for which we did not find an witness,

\[ \mathbb{E}(X) = \mathbb{E} \left( \sum X_{ij} \right) = \sum \mathbb{E}(X_{ij}) \leq n^2/n^3 = 1/n \]

The running time for the above algorithm is

\[ = n^w \cdot \log^2 n + \frac{1}{n} n^2 \]
4 Deterministic Algorithm for BPWM

4.1 Some Background

**Definition 2.** Let $n$ and $k$ be fixed positive integers. A collection $\mathcal{F}$ of $k$-sets of $[n]$ is a completely separating system if, for all distinct $X_1, X_2 \in [n]$, there is an $S \in \mathcal{F}$ for which $X_1 \in S$ and $X_2 \notin S$.

Let $X = X_1 \cup X_2$ and $|X| = x$.

Call $S$ good for this $X$ if above condition holds. $\mathcal{F}$ is also called $(n,k,x)$ seperating set. Let $l$ denote the minimum size of such a $\mathcal{F}$.

Suppose, we know $|P_{ij}| \leq k$, then we want to find $(n,1,k)$ seperating collection so that we can reduce the problem to unique witness finding problem. let us find the minimum value for $l$ in this case.

$$\Pr[\text{No } S \text{ is good for some } X] = (1 - \frac{1}{2^k})^l$$

$$\Pr[\text{No } S \text{ is good for all } x] = (1 - \frac{1}{2^k})^{l \cdot \binom{n}{k}} 2^k \text{ [by Union Bound]}$$

Therefore if some good set exists,

$$(1 - \frac{1}{2^k})^l \cdot \binom{n}{k} 2^k < 1$$

$$e^{-l/2} \cdot n^k 2^k < 1$$

$$(2n)^k < e^{l/2}$$

$$k \log n < \frac{l}{2k}$$

$$l \geq 2^k k \log n$$

4.2 Algorithm For BPWM

Define $c = \lceil \log \log n + 9 \rceil$ and $\alpha = \frac{8}{2^c}$. For two matrices $C$ and $R$ with $\{0, 1\}$ entries define $Y = C \land R$ by $Y_{ij} = C_{ij} \land R_{ij}$.

The algorithm is as follows. Besides $A$, $B$ and $C = AB$ it uses two additional matrices: $R$ and $D$.

The way to perform steps 5 and 6 are described later.

**while not all witnesses are known do**

1. Let $L$ denote the set of all positive entries of $C$ for which there are no known witnesses.
2. Let $R$ be the all 1 matrix.
3. repeat
4. $D \leftarrow A(B \land R)$ (The matrix multiplication is over the integers)
5. Let $L'$ denote the set of all entries of $D$ in $L$ which are at most $c$.
6. Find witnesses for all entries in $L'$.
7. $R \leftarrow$ good matrix (see definition of good below).
8. until $1 + 3 \log_{4/3} n$ times ;
9. end

**Algorithm 3:** Find Witness

A matrix $R$ is good (in step above) if the following two conditions hold:
(i) The total sum of the entries of \( D = A(B \land R) \) in \( L \) is at most \( 3/4 \) of what this sum was while using the previous matrix \( R \).

(ii) The fraction of entries of \( D \) in \( L \) that go from a value bigger than \( c \) to 0 is at most \( \alpha \).

**Lemma 3.** If \( R \leftarrow R \land S \) in step 6 where \( S \) is a random \( \{0, 1\} \) matrix, then the new \( R \) is good with probability at least \( 1/6 \).

**Proof.** The lemma follows from the following three claims:

**Claim 4.** \( \Pr[\text{the sum of entries of } D \text{ in } L \leq \frac{3}{4}W] \geq 1/3 \).

**Proof.** Let \( X = \text{new sum of entries of } D \text{ in } L \).

\[
\mathbb{E}[X] = \frac{\text{previous sum of entries of } D \text{ in } L \times 2}{2} = \frac{W}{2}
\]

\[
\Pr[X > \frac{3}{4}W] \leq \frac{\mathbb{E}(X)}{\frac{3}{4}W} = \frac{W/2}{\frac{3}{4}W} = 2/3 \text{ [Using Markov’s Inequality]}
\]

Therefore, \( \Pr[\text{the sum of entries of } D \text{ in } L \leq \frac{3}{4}W] \geq 1/3 \).

**Claim 5.** The probability that a fixed entry of \( D \) which is at least \( c \) drops down to 0 \( \leq \frac{1}{2^c} \).

**Proof.** If a entry in the integer multiplicatin \( c \) then were \( c \) 1s, summing up to give \( c \). All of these \( c \) entries should become 0. Each entry goes to 0 with probability \( 1/2 \).

If we assume that entries of \( S \) are \( c \)-wise independent,

\( \Pr[\text{all of the } c \text{ entries goes to 0}] = 1/2^c \).

Therefore the probability that a fixed entry of \( D \) which is at least \( c \) drops down to 0 \( \leq \frac{1}{2^c} \).

**Claim 6.** The probability that more than a fraction \( \alpha \) of the entries of \( D \) in \( L \) drop from at least \( c \) to 0 is at most \( \frac{1}{8} \).

**Proof.** Let \( T \) be the no of values which are greater than \( c \) that goes to 0. and \( n_{\geq c} \) be the values garter than \( c \).

\[
\Pr[T \geq \alpha n_{\geq c}] \leq \frac{\mathbb{E}(T)}{\alpha \cdot n_{\geq c}} = \frac{1/2^c \cdot n_{\geq c}}{\alpha \cdot n_{\geq c}} \quad \text{[From claim 4]}
\]

\[
= \frac{1}{2^c} \cdot \frac{1}{\alpha} = 1/8
\]

Therefore \( \Pr[R \text{ is good}] = 1/3 - 1/8 > 1/6 \).
4.3 c-wise $\epsilon$-dependent \{0,1\} random variables

Define $\epsilon = 2c + 1$. The crucial point is to observe that the proof of the above lemma still holds, with almost no change, if the matrix $S$ is not totally random but its entries are chosen from a $c$-wise $\epsilon$-dependent distribution. Recall that if in random variables whose range is \{0, 1\} are $c$-wise $\epsilon$-dependent then every subset of $i \leq c$ of them attains each of the possible $2^i$ configurations of 0 and 1 with probability that deviates from $1/2^i$ by at most $\epsilon$.

**Lemma 7.** If $R \leftarrow R \land S$ in step 6 where the entries of $S$ are chosen as $n^2$ random variables that are $c$-wise $\epsilon$-dependent, then the new $R$ is good with probability at least $1/12 - 2\epsilon$

We note that in fact it is sufficient to choose only one column and copy it $n$ times. The proof is by the following modified three claims, whose proof is analogous to that of the corresponding previous ones.

**Claim 8.** The probability that the sum of entries of $D$ in $L$ goes down by at least a factor of $3/4$ is at least $1/3 - 2\epsilon$

**Claim 9.** The probability that a fixed entry of $D$ which is at least $c$ drops down to 0 is at most $1/2^c + \epsilon$

**Claim 10.** The probability that more than a fraction $\alpha$ of the entries of $D$ in $L$ drop from at least $c$ to 0 is at most $1/4$

**Proof.**

Pr[a entry with value $c$ in $D$ goes to 0] = $1/2^c + \epsilon$.

Pr[a entry with value at least $c$ in $D$ goes to 0] $\leq (\frac{1}{2^c} + \epsilon)$

Pr[more than $\alpha$ fraction drop from at least $c$ to 0] $\leq (\frac{1}{2^c} + \epsilon) \frac{1}{\alpha} < \frac{2^c}{2^c \alpha} = 1/4$.

Thereby Pr[R is good] = $1/3 - 2\epsilon - 1/4 = 1/12 - 2\epsilon$.

Think of each entries of $R$ as a random variable. Therefore we have $n^2$ random variables. As shown in [1] and [2] there are explicit probability spaces with $n^2$ random variables which are $c$-wise $\epsilon$-dependent, whose size is

$$(\log n.c. \frac{1}{\epsilon})^{2+o(1)}$$

If we take $\epsilon = \frac{1}{2^c + 1}$. No of such $R$ we can have is

$$(\log n.c.2^{c+1})^{2+o(1)} = poly \log n$$

Now suppose that in step 6 all the matrices $S$ defined by such a probability space are searched, until a good one is found.

Note that during the performance of step 6, while considering all possible matrices $S$ provided by our distribution, we can accomplish step 5 as well. This is true since $c$-wise $\epsilon$-dependence guarantees that every entry in $L$ will drop to precisely 1 for some of the matrices $S$ and hence, if we replace each matrix multiplication in the search for a good $S$ by two matrix multiplications, we complete steps 5 and 6 together.
4.4 Running Time:

Checking whether a matrix is good requires only matrix multiplication plus $O(n^2)$ operations. Therefore the inner loop (starting at step 3) takes $(\text{polylog } n)O(n^w)$. In every iteration of the inner loop starting at 3 at most $\alpha$ fraction of the entries of $L$ are thrown (i.e. their witness will not be found in this iteration of the outer loop).

Therefore at least $(1 - \alpha)^{1 + 3\log 4/3n}$ fraction of the entries of $D$ in $L$ will not be thrown during the completion of these iterations. For those entries, which are at least $1/2$ of the entries in $L$, a witness is found. Therefore, only $O(\log n)$ iterations of the outer loop are required, implying the desired $O(n^w(\log n)^{O(1)})= \tilde{O}(n^w)$ total running time.

References

