Trace Reconstruction

Professor Michel Pain, William Lu

June 2, 2020

Contents

1 Introduction 3
1.1 Concept . . . 3
1.2 Goals and Applications . . . 3

2 An Exponential Trace Algorithm for Random X and Small q. 3
2.1 Basic ideas . . . 3
2.2 Different Cases . . . 4
2.2.1 When i Is Equal to 1 . . . 4
2.2.2 When i Is Equal to 2 or Above . . . 4
2.3 What Precision Do We Need? . . . 5
2.4 Number of Traces Required . . . 6
2.5 New Case: When i Is Large but Still Smaller than Some Number h . . . 7

3 Polynomial Traces, Random X 8
3.1 What Precision Do We Need? . . . 12
3.2 Number of Traces Required . . . 12
3.3 Unknown Variables in S . . . 13

4 Algorithm 15

1 Introduction

1.1 Concept

The topic of our research project is trace reconstruction. Consider that there is a random binary sequence. Every time there will be a new trace being generated when this original random binary string is transmitted. Each bit of the original binary string will face a fixed probability of being erased, due to various factors that people may not be able to control. In other words, every trace is obtained when the binary string is passing through a "deleting channel," which extirpates bits arbitrarily with a probability q. Hence, traces are distinct subsequences of the original binary string/sequence. Here, we can call the original sequence X and the new trace Y.

1.2 Goals and Applications

The goal of our project is to figure out how many traces we need at least to reconstruct the original sequence of length n with high probability.

For this, our first aim will be to understand the results obtained by Holenstein et al in Trace reconstruction with constant deletion probability and related results. They prove different upper bounds on the number of traces required for the reconstruction, with two phases depending on the deletion probability q: for small q, only a polynomial number of traces are required, whereas for large q, the number is exponential in $n^{1/2}$ At the end, we will also implement the algorithm into a Python program. Beyond completing the original message, trace reconstruction has a lot of other applications in a wide range of subject areas.

For instance, trace reconstruction can be applied on DNA when we try to find out the common ancestors from available traces. We can also use the concept of trace reconstruction in the context of sensor network, since it is frequent that the sensor cannot detect all the event (bits). However, reconstructing each events will allow us to be exempt from the noise and imperfections of the sensors.

2 An Exponential Trace Algorithm for Random X and Small q.

2.1 Basic ideas

Consider that the original and complete string X consists of n bits. That is, X can be written as x_1, x_2, x_3 ... x_{n-1}, x_n, where each bit can be the value of either 1 or 0. Now, if we want to find out the probability of i-th bit – where i is between 1 and n – ends up at j-th bit in the first trace Y that we receive after X passing through the deletion channel, we can write the probability equation in binomial form.

$$P(i,j) = \binom{i-1}{j-1} \cdot q^{i-j} \cdot (1-q)^j$$

We write the probability in binomial form since each bit i has probability q being deleted, and j is always smaller than or equal to i. There are only $j-1$ bits out of $i-1$ bits not deleted after the string going through the deletion channel. We keep the dependence on q implicit throughout. Besides, we can write the probability of Y_j as the sum of the product of the probability $P(i,j)$ and x_i. This is also the expectation of Y_j, due to the fact that x_i being the indicator random variable.

$$P(Y_j) = \sum_{i \geqslant j} P(i,j) \cdot x_i = E[Y_j]$$

2.2 Different Cases

The calculation of probability for i being small and large is very different. In the paper by Holenstein et al, the difference between the two cases is not specified. Here, we consider the problem in two cases. First, when i is small. For example, i is equal to 1 and 2. And when i is a large number but smaller than some number h. $(x_1...x_i...x_h......x_n)$

2.2.1 When i Is Equal to 1

When we want to know if bit x_i is remained or not (If Y_1 is equal to x_1), the only solution to find out this is by collecting enough data, recording the number of times that Y_1 is 1 and Y_1 is 0. Because of the law of large number, we can know what the value is of Y_1. Keep the idea that q is small.

$$P(Y_1 = x_1) \geqslant 1 - q$$

$$P(Y_1 \neq x_1) \leqslant q$$

In conclusion, we estimate $P(Y_1 = 1)$. If the number of times $Y_1 = 1$ appears more than half of all the times, $x_1 = 1$. Else, $x_1 = 0$.

2.2.2 When i Is Equal to 2 or Above

In the case of i being 2, bit x_2 could either end up at Y_1 or at Y_2. If bit x_2 ends up at Y_2, it implies that x_1 ends up at Y_1. If bit x_2 ends up at Y_1, it implies that x_1 is deleted. To calculate the probability, we need to go back to the case of Y_1.

$$P(Y_1 = 1) = (1-q) \cdot x_1 + q \cdot (1-q) \cdot x_2 + q \cdot q \cdot (1-q) \cdot x_3 + ... \tag{1}$$

With a known q, like $1/3$, we can know the initial bits of the original string. Otherwise, the sum will not follow the recording result. To be more specific, if we want to find the value of x_3, here is an example.

Example 2.1. Assuming $P(Y_1 = 1) = 0.7378$ and q $= 1/3$. Then x_1 has to be 1 as its contribution is approximately 0.66. The contribution of x_2 is $0.33 \cdot 0.66 \approx 0.2178$, then x_2 has to be 0, otherwise the equation is unsatisfied. The contribution of x_3 is $0.33 \cdot 0.33 \cdot 0.66 \approx 0.07184$, then x_3 has to be 1 and so on. The precision of x_3 here would then be 0.0179.

To accomplish our goal: finding x_i, we shall ask ourselves two questions. What precision do we need for $P(Y_1 = 1)$? How many traces do we need to get this precision?

2.3 What Precision Do We Need?

Now, before answering the two questions, We can write the probability of Y_1 being equal to 1 in a generalized form as below:

$$P(Y_1 = 1) = \sum_{j=1}^{n} x_j \cdot q^{j-1} \cdot (1-q) \tag{2}$$

If we want to find x_i, in the first step, we move equation (2) around like this.

$$x_i \cdot q^{i-1} \cdot (1-q) = P(Y_1 = 1) - \sum_{j=1}^{i-1} x_j \cdot q^{j-1} \cdot (1-q) - \sum_{j=i+1}^{n} x_j \cdot q^{j-1} \cdot (1-q)$$

Denote s

$$s = P(Y_1 = 1) - \sum_{j=1}^{i-1} x_j \cdot q^{j-1} \cdot (1-q)$$

We are able to estimate s since we already know x_1,x_2,..x_{i-1}. s can also be written as

$$s = x_i \cdot (1-q) \cdot q^{i-1} + \sum_{j=i+1}^{n} x_j \cdot q^{j-1} \cdot (1-q)$$

Follow equation (1) from the above. The sum after the first three terms is $\sum_{n \geqslant 3} q^n \cdot (1-q) = q^3 \cdot (1-q) \cdot \sum_{k \geqslant 0} q^k = q^3 \leqslant q^2/3$, which is $\leqslant \frac{q^2 \cdot (1-q)}{2}$. That is, the sum is smaller than half of the previous contribution of x_3. Now, with the same logic, for x_i, the sum after the first i terms will be smaller than half of the contribution of x_{i-1}. Namely, $\frac{q^{i-1} \cdot (1-q)}{2}$. However, due to the existence of possible errors, we need to set the precision to be $\frac{q^{i-1} \cdot (1-q)}{4}$ for any i. If the probability is larger than the sum of precision and the contribution of x_i, x_i will be 1. If not, we let x_i to be 0.

We have shown that $\sum_{j=i+1}^{n} x_j \cdot q^{j-1} \cdot (1-q)$ will be smaller than $\frac{(1-q) \cdot q^{i-1}}{2}$, so if $s \geqslant (1-q) \cdot q^{i-1}$, $x_i = 1$. If $s \leqslant \frac{(1-q) \cdot q^{i-1}}{2}$, then $x_i = 0$. We let our bounds to be $\frac{3(1-q) \cdot q^{i-1}}{4}$ to include the consideration of discrepancies. That is, if s is smaller than $\frac{3(1-q) \cdot q^{i-1}}{4}$, then $x_i = 0$. If s is greater than or equal to $\frac{3(1-q) \cdot q^{i-1}}{4}$, then x_i will be 1.

In this way, we can reconstruct i bits of the original string just by getting the probability of Y_1 equal to 1 and knowing what x_1 is. For example, to find x_2, we first calculate $s = P(Y_1 = 1) - (1-q) \cdot x_1$, then we check if s is smaller than $\frac{3(1-q) \cdot q}{4}$. If yes, then x_2 is 0. Otherwise, x_2 is 1. We can do this many times and recover the original sequence to bit i.

Remark 2.2. It might seem intuitive to calculate the probability of Y_2 being 1 in order to find the value for X_2. However, we cannot do the same with finding the probability of Y_2 equal to 1 as q = 1/3. Because $P(Y_2 = 1) \geqslant (1-q)^2 \geqslant 4/9$. $P(Y_2 = 0) \leqslant 5/9$, there is no way to distinguish if $P(Y_2 = 1) \in [4/9, 5/9]$. But we can find $P(Y_2 = 1)$ if q is small and write the probability equation as

$$\begin{aligned} P(Y_2 = 1) = E[Y_2] = E[\sum_{j=2}^{n} x_j \cdot I_{Fj}] = \sum_{j=2}^{n} x_j \cdot E[I_{Fj}] = \sum_{j=2}^{n} x_j \cdot P(F_j) \\ = (1-q)^2 \cdot x_2 + 2q \cdot (1-q)^2 \cdot x_3 + ... \end{aligned}$$

2.4 Number of Traces Required

Getting back to the question of how many traces we require to attain the precision, we apply Hoeffding's inequality $P(|\bar{Y_1} - E[\bar{Y_1}]| > t) \leqslant 2e^{-2Nt^2}$, where $\bar{Y_1}$ is $\frac{{Y_1}^1 + {Y_1}^2 + ... + {Y_1}^M}{M}$. Y^1 represents the first trace we receive, and it can be written as $({Y_1}^1,\ {Y_2}^1 ... {Y_n}^1)$. If $Y^1 ... Y^M$ are identically distributed, then $E[\bar{Y_1}] = E[Y_1]$ with the same distribution as X. Define $P(Y_1 = 1)$ as p, and thus $p = E[Y_1]$. Let precision be our $t = \frac{(1-q) \cdot q^{i-1}}{4}$. Therefore, we can estimate N (the number of traces that we need) by specifying a good bound and do the same for $P(Y_j = 1)$ as well. If our precision of $P(Y_j = 1)$ is precise enough with probability at least $1 - \frac{1}{n^{1+d}}$, then the estimations for all $P(Y_j = 1)$ are precise enough with probability at least $1 - \frac{1}{n^d}$. In other words, P(one given estimation is bad) $\leqslant \frac{1}{n^{1+d}}$, then P(at least one of the estimations is bad) $\leqslant n \cdot \frac{1}{n^{1+d}} = \frac{1}{n^d}$

Here, we want $e^{-2Nt^2} \leqslant \frac{1}{n^{1+d}}$, where $t = \frac{(1-q) \cdot q^{i-1}}{4}$.

$$\begin{aligned} -2Nt^2 &\leqslant ln(\frac{1}{n^{1+d}}) \\ N &\geqslant \frac{ln(\frac{1}{n^{1+d}})}{-2t^2} \\ &= \frac{8(1+d)ln(n)}{(1-q)^2 q^{2(i-1)}} \end{aligned}$$

In conclusion, we need at least $N = \frac{8(1+d)ln(n)}{(1-q)^2 q^{2(i-1)}}$ traces to estimate $P(Y_1 = 1)$ with a precision good enough to find x_i. (i to be n)

2.5 New Case: When i Is Large but Still Smaller than Some Number h

As i is large, x_i will end up in position $j \leqslant (1-q)\cdot i$. But x_{i+1} has approximately the same probability to end up at $(1-q)\cdot i$ is the same as x_i. Thus, if we alter the equation to $j \leqslant (1-3q)\cdot i + 3q$, then we can calculate the probability for i ends up at j.

For any **a** larger than **i**, we can get:

$$\frac{P(a,j)}{P(a-1,j)} = \frac{\binom{a-1}{j-1}}{\binom{a-2}{j-1}} \cdot q = \frac{a-1}{a-j} \cdot q \leqslant 1/3$$

When $(1-4q)\cdot i + 4q \leqslant j \leqslant (1-3q)\cdot i + 3q$ then

$$\begin{aligned} P(i,j) &= \binom{i-1}{j-1} \cdot q^{i-j} \cdot (1-q)^j \geqslant (\frac{i-1}{i-j})^{i-j} \cdot q^{i-j} \cdot (1-q)^j \\ &\geqslant (\frac{1}{4q})^{i-j} \cdot q^{i-j} \cdot (1-q)^j \geqslant (\frac{1}{4})^{i-j} \cdot (1-q)^j \\ &\geqslant (\frac{1}{4})^{3qi} \cdot e^{ln(1-q)\cdot(1-4q)\cdot i} \geqslant e^{-6iq} \end{aligned}$$

Lemma 2.3. *If $j \leqslant (1-3q)\cdot i + 3q$, then $P(i,j) \geqslant 2\sum_{i'>i} P(i',j)$. If $(1-4q)\cdot i + 4q < j < (1-3q)\cdot i + 3q$, then $P(i,j) \geqslant e^{-6iq}$*

And also, we can write the probability of Y_j being equal to 1 as the following:

$$P(Y_j = 1) = \sum_{l=j}^{n} P(l,j)\cdot x_l = \sum_{l=j}^{i-1} P(l,j)\cdot x_l + P(i,j)\cdot x_i + \sum_{l=i+1}^{n} P(l,j)\cdot x_l$$

We can move the equation around like before, with known variables on one side.

$$P(i,j)\cdot x_i = P(Y_j = 1) - \sum_{l=j}^{i-1} P(l,j)\cdot x_l - \sum_{l=i+1}^{n} P(l,j)\cdot x_l$$

Denote r

$$r = P(Y_j = 1) - \sum_{l=j}^{i-1} P(l,j)\cdot x_l$$

We are able to do this because we already know $x_1 ... x_{i-1}$. And we know $P(i,j) = \binom{i-1}{j-1}\cdot q^{i-j}\cdot(1-q)^j$. So r can also be written as

$$r = \binom{i-1}{j-1}\cdot q^{i-j}\cdot(1-q)^j \cdot x_i + \sum_{l=i+1}^{n} P(l,j)\cdot x_l$$

Since $\sum_{l=i+1}^{n} P(l,j)\cdot x_l$ will be smaller than half of the contribution of $P(i,j)$, it is enough to have a precision that is $\frac{P(i,j)}{4} = \frac{e^{-6qi}}{4}$ to estimate $P(Y_j = 1)$. To include discrepancies, if $r \geqslant \frac{3P(i,j)}{4} = \frac{3e^{-6qi}}{4}$, then $x_i = 1$. If $r < \frac{3P(i,j)}{4} = \frac{3e^{-6qi}}{4}$, then x_i will be 0.

To determine the number of traces required to reconstruct the string, we want $e^{-2Nt^2} \leqslant \frac{1}{n^{1+d}}$, where $t = \frac{e^{-6qi}}{4}$. (From above)

$$\begin{aligned} -2Nt^2 &\leqslant ln(\frac{1}{n^{1+d}}) \\ N &\geqslant \frac{ln(\frac{1}{n^{1+d}})}{2t^2} \\ &= \frac{8(1+d)ln(n)}{e^{-12qi}} \end{aligned}$$

In conclusion, we need at least $N = \frac{8(1+d)ln(n)}{e^{-12qi}}$ traces to estimate $P(Y_j = 1)$ with a precision good enough to find x_i.

3 Polynomial Traces, Random X

After we have derived the exponential trace algorithm with a known length of X, we now consider the case when X is a uniform random string by using polynomial traces. As we have learned $x_1x_2...x_{i-1}$, we define the substring S as $x_{i-w}x_{i-w+1}...x_{i-1}$ of width w. If substring S is contained in the trace Y that we receive, matching to a substring $y_{j-w}...y_{j-1}$, then we may be able to find out x_i based on y_j. Nevertheless, there's a proportion of $1-(1-q)^w$ traces which cannot be used due to their lack of the complete substring. Also, suppose S is something like $(...0, 0, 0)$ and x_i is 0. Though the substrings in X and in the trace might match, we cannot ensure that one of these three zeros is not substituted by x_i. We cannot notice the error.

Definition 3.1. We call a string X of length n is w-substring unique if for any a, b, one of the following holds:

1. $b \leqslant a$ or $b + 1.1w \geqslant a + w$
2. $X(a : a + w)$ cannot be obtained by deleting some symbols in $X(b : b + 1.1w)$

Lemma 3.2. *At least a fraction of* $1 - \frac{1}{n^d}$ *all strings of length* n *are* $(6 + 3d)log(n)$*-substring unique.*

Proof. Let's consider the case where only $b \leqslant a$ or $b+1.1w \geqslant a+w$ happens. In the case of only $b + 1.1w \geqslant a + w$ happening, the probability that $X(a : a + w)$ can be obtained by deleting bits in $X(b : b+1.1w)$ is approximately $\binom{1.1w}{0.1w} \cdot (1/2)^w$, because there is $0.1w$ choice of subsets of entries of $X(b : b+1.1w)$ that will be deleted. To find the boundary for this probability, we use Stirling's formula, $\sqrt{2\pi}n^{n+1/2}e^{-n} \leqslant n! \leqslant en^{n+1/2}e^{-n}$. Let k be $0.1w$. Take the left part of the inequality as in k and w, the right part of the inequality as in $(k+w)$.

$$
\begin{aligned}
\binom{1.1w}{0.1w} \cdot (1/2)^w = \binom{k+w}{k} \cdot (1/2)^w &= \frac{(k+w)!}{k!w!} \cdot (1/2)^w \\
&\leqslant \frac{e}{\sqrt{2\pi}} \cdot \frac{\sqrt{2\pi(k+w)} \cdot (\frac{k+w}{e})^{k+w}}{\sqrt{2\pi(k)}(\frac{k}{e})^k \cdot \sqrt{2\pi(w)}(\frac{w}{e})^w} \cdot (1/2)^w \\
&= \frac{e}{\sqrt{2\pi}} \cdot \frac{(k+w)^{k+w} \cdot \sqrt{k+w} \cdot (\frac{1}{e})^{k+w}}{\sqrt{2\pi kw} \cdot k^k \cdot w^w \cdot (\frac{1}{e})^{k+w}} \cdot (1/2)^w \\
&= \frac{e}{\sqrt{2\pi}} \cdot \frac{(k+w)^{k+w}}{k^k \cdot w^w} \cdot \frac{1}{\sqrt{2\pi}} \cdot \sqrt{\frac{k+w}{kw}} \cdot (1/2)^w \\
&= \frac{e}{\sqrt{2\pi}} \cdot \frac{1}{\sqrt{2\pi}} \cdot \frac{(1.1w)^w \cdot (1.1w)^{0.1w}}{(0.1w)^{0.1w} \cdot w^w} \cdot \sqrt{\frac{11}{w}} \cdot (1/2)^w \\
&= \frac{e}{\sqrt{2\pi}} \cdot \frac{1}{\sqrt{2\pi}} \cdot \sqrt{\frac{11}{w}} \cdot (1.1/2)^w \cdot 11^{0.1w} \\
&= \frac{e}{2\pi} \cdot \frac{1}{\sqrt{w}} \cdot \sqrt{11} \cdot (\frac{1.1 \cdot 11^{0.1}}{2})^w \leqslant 0.7^w
\end{aligned}
$$

if w is greater than 3.

Now, since there are n choices for both a and b to be, there must exist less than n^2 disjoint intervals. The probability of X is not w-substring unique will be smaller than or equal to $n^2(0.7)^w$. Our goal is to find the value of a when $n^2(0.7)^w \leqslant n^{-d}$ and $w = a \cdot log(n)$.

$$n^{2+log(0.7)a+d} \leqslant n^0$$

$$2 + log(0.7)a + d \leqslant 0$$

$$a \geqslant \frac{-(2+d)}{log(0.7)} \simeq 2.8(2+d)$$

Hence, when $a = 6 + 3d$, the probability function holds true. □

Lemma 3.3. *Let q be a small enough constant and let X of length n be w-substring unique. Let Y be a trace after X passing through a deletion channel with deletion probability q. If Y contains a string $Y(j-w:j)$ matches $X(i-w:i)$, then the probability of y_{j-1} doesn't come from a bit in the range $x_{i-1}..x_{i-1+0.1w}$ is at most this: $\frac{n \cdot exp(-2.2w \cdot (\frac{0.1-1.1q}{1.1})^2)}{(1-q)^w}$. And we denote it by γ.*

Proof. There are two different cases to consider in mind. Either for any a, no more than $0.1w$ bits are deleted in $X(a:a+1.1w)$, or there exists a fixed a such that more than $0.1w$ bits are deleted in $X(a:a+1.1w)$.

Assume that y_{j-1} comes from $x_{a+1.1w-1}$. In the first scenario — no more than $0.1w$ entries are deleted — y_{j-w} must come from $x_a...x_{a+0.1w}$, and $X(i-w:i)$ is included in

$X(a : a+1.1w)$ as X is w-substring unique. Therefore, $a \leqslant i-w \leqslant i \leqslant 1.1w$, which means $a+w \leqslant i \leqslant a+1.1w$ and $i-1 \leqslant a+1.1w-1 \leqslant i+0.1w-1$. Now, we can say that y_{j-1} comes from $x_{a+1.1w-1}$ or from the bits before.

In the second scenario, we find some a and denote Z to be the number of bits deleted in the substring $X(a : a+1.1w)$. $E[Z] = q \cdot 1.1w$.

$$P(Z > 0.1w) = P(Z - E[Z] > (0.1 - q \cdot 1.1) \cdot w)$$

By using Hoeffding's inequality, we get $P(\frac{Z}{1.1w} - E[\frac{Z}{1.1w}] > t) = P(Z - E[Z] > 1.1wt) \leqslant e^{-2 \cdot 1.1w \cdot t^2}$. Then, $P(Z > 0.1w) \leqslant exp(-2.2w \cdot (\frac{0.1-1.1q}{1.1})^2)$. , where $0.1 - q \cdot 1.1$ has to be greater than 0, so q has to be smaller than $\frac{1}{11}$. Then, P(for any a, more than 0.1w are deleted) $\leqslant n \cdot exp(-2.2w \cdot (\frac{0.1-1.1q}{1.1})^2)$

Remark 3.4. When $q \geqslant 0.0132$, this probability is greater than 1. The lemma is then pointless.

Conditioned on the event that Y contains a substring $Y(j-w : j)$ that matches $X(i-w : i)$, and we call it event G. We let event E be the event that y_{j-1} comes from $x_{i-1}...x_{i-1+0.1w}$. Event F be the event that there exists an "a" such that more than 0.1w entries in $X(a : a+1.1w)$ are deleted. Then $E \subset F$. The probability of y_{j-1} coming from $x_{i-1}...x_{i-1+0.1w}$ given that $Y(j-w : j)$ matching $X(i-w : i)$ is:

$$P(E|G) = \frac{P(E \cap G)}{P(G)} \leqslant \frac{P(E)}{P(G)} \leqslant \frac{P(F)}{P(G)} = \frac{n \cdot exp(-2.2w \cdot (\frac{0.1-1.1q}{1.1})^2)}{(1-q)^w}$$

□

Theorem 3.5. *Let q be a small enough constant and X be 100log(n)substring unique. With lemma 3.1 and lemma 3.2, we can get a polynomial time algorithm that reconstructs X with probability at least $P(E) \geqslant 1-\gamma$ or in $1-o(1)$ from poly(n) independent traces of X.*

Let q be a small enough constant and assume that we have obtained $x_1...x_{i-1}$. For each trace, if it contains substring $x_{i-v-w+1}...x_{i-v}$, then we discard all the bits before x_{i-v}, denoting the remainder of the trace as Y^{new}. $(v = \frac{w}{q})$. Let R be the random variable which represents the position of the last bit in the original substring. In other words, Given $R = r$, Y^{new} is a trace of $x_{r+1}...x_n$. $\{i-v \leqslant R \leqslant i-v+0.1w\} = E$

$$P(Y_j^{new} = 1) = P(\{Y_j^{new} = 1\} \cap E) + P(\{Y_j^{new} = 1\} \cap E^c)$$

Let $P(\{Y_j = 1\} \cap E^c)$ be $A_i(X)$, then by lemma 3.2, we know that $0 \leqslant A_i(X) \leqslant$

$P(E^c) \leqslant \frac{n \cdot exp(-2.2w \cdot (\frac{0.1-1.1q}{1.1})^2)}{(1-q)^w} = \gamma$

$$\begin{aligned}
P(Y_j^{new}=1) &= \sum_{r=i-v}^{i-v+0.1w} P(\{Y_j^{new}=1\} \cap E | R=r) \cdot P(R=r) + A_i(X) \\
&= \sum_{r=i-v}^{i-v+0.1w} P(Y_j^{new}=1 | R=r) \cdot P(R=r) + A_i(X) \\
&= \sum_{r=i-v}^{i-v+0.1w} P(R=r) \sum_{l=1}^{n} P(l,j) x_{r+l} + A_i(X) \\
&= \sum_{r=i-v}^{i-v+0.1w} P(R=r) \sum_{l=r+1}^{n} P(l-r,j) x_l + A_i(X)
\end{aligned}$$

Because we know $x_1 ... x_{i-1}$,

$$P(Y_j^{new}=1) = \sum_{r=i-v}^{i-v+0.1w} P(R=r) \left(\sum_{l=r+1}^{i-1} P(l-r,j) x_l + P(i-r,j) x_i + \sum_{l=i+1}^{n} P(l-r,j) x_l \right) + A_i(X) \tag{3}$$

$F = \{Y$ contains the substring $\}$. Let $\sum_{r=i-v}^{i-v+0.1w} P(R=r|F) \sum_{l=r+1}^{i-1} P(l-r,j) x_l$ be S. And by lemma 2.3 , due to $j \leqslant (v-0.1w)(1-3q) \leqslant (i-r)(1-3q)$ for all r, we know that $\sum_{i+1}^{n} P(l-r,j) x_l \leqslant \frac{1}{2} P(i-r,j)$.

So when $x_i = 0$, from equation (3), we can write:

$$P(Y_j^{new}=1) \leqslant A_i(X) + S + \sum_{r=i-v}^{i-v+0.1w} P(R=r) \frac{1}{2} P(i-r,j) \tag{4}$$

When $x_i = 1$, from equation (3), we can write:

$$P(Y_j^{new}=1) \geqslant A_i(X) + S + \sum_{r=i-v}^{i-v+0.1w} P(R=r) P(i-r,j) \tag{5}$$

Since $0 \leqslant A_i(X) \leqslant \gamma$, we can see that the largest difference between $A_i(X)$ is at most γ. Since we know $x_1 ... x_{i-1}$, S can be calculated with a precision $t = (\frac{1}{2})^{4w}$ in polynomial time. Due to $j \leqslant (v-0.1w)(1-3q) \leqslant (i-r)(1-3q)$ for all r:

$$P(i-r,j) \geqslant (\frac{1}{2})^{-(i-r-j)} \geqslant (\frac{1}{2})^{-(v-j)} = (\frac{1}{2})^{0.1w+3qv-0.3qw} = (\frac{1}{2})^{0.1w+3w-0.3qw} \geqslant (\frac{1}{2})^{4w}$$

Thus, the gap between (4) and (5) is at least $(\frac{1}{2})^{4w+1}(1-\gamma)-\gamma$. In the worst case for the gap, the threshold will be in the middle of (4) and (5): $\frac{\gamma}{2} + S + \frac{3}{4} \sum_r P(R=r) P(i-r,j)$.

3.1 What Precision Do We Need?

Since $P(E) \geqslant \gamma$, if we assume that $\gamma \leqslant (\frac{1}{2})^{4w+2}$, the difference between the two scenarios is at least

$$\begin{aligned}(1-\gamma)\cdot(\frac{1}{2})^{4w+1} - \gamma &\geqslant (\frac{1}{2})^{4w+1} - (\frac{1}{2})^{8w+3} - (\frac{1}{2})^{4w+2} \\ &= (\frac{1}{2})^{4w+2} - (\frac{1}{2})^{8w+3} \\ &= (\frac{1}{2})^{4w-2}\cdot\left(1-(\frac{1}{2})^{4w+1}\right) \geqslant (\frac{1}{2})^{4w+3}\end{aligned}$$

, which is the smallest possible gap. Therefore, with one fourth of the gap, a precision less than $(\frac{1}{2})^{4w+5}$ allow us to estimate $P(Y_j^{new}=1)$.

3.2 Number of Traces Required

But how many traces do we need to get a precision t with probability $1-\frac{1}{n^{1+d}}$? We know that $P(Y^{new}$ contains the substring $x_{i-v-w+1}...x_{i-v})$ is $\geqslant (1-q)^w$.

How many traces containing the substring do we need? If our precision of $P(Y_j^{new} = 1)$ is precise enough with probability at least $1-\frac{1}{2n^{1+d}}$, then P(one given estimation is bad) $\leqslant \frac{1}{2n^{1+d}}$. Using Hoeffding's inequality, $P(|\bar{Y}^{new} - E[\bar{Y}^{new}]| > t) \leqslant 2e^{-2N_1t^2}$, we want $e^{-2Nt^2} \leqslant \frac{1}{2n^{1+d}}$, where $t = \frac{1}{2}^{4w+4}$. (From above)

$$\begin{aligned}-2N_1t^2 &\leqslant ln(\frac{1}{2n^{1+d}}) \\ N_1 &\geqslant \frac{ln(\frac{1}{2n^{1+d}})}{-2t^2} \\ &= \frac{ln(\frac{1}{2}) - ln(n)(1+d)}{-(\frac{1}{2})^{8w+7}} \\ &= (ln(n)(1+d) + ln(2))2^{8w+7}\end{aligned}$$

How many traces do we need to get at least N_1 traces containing the substring with probability $\geqslant 1-\frac{1}{2n^{1+d}}$? Let Z_i be 1, if the i-th trace contains the substring; Z_i be 0, otherwise. $\bar{Z} = \frac{\sum_1^N Z_i}{N}$.

$$P(N\bar{Z} < N_1) = P(\bar{Z} < \frac{N_1}{N}) = P(\bar{Z} - E[\bar{Z}] < \frac{N_1}{N} - E[\bar{Z}]) \leqslant e^{-2N(E[\bar{Z}]-\frac{N_1}{N})^2}$$

Now first assume $\frac{N_1}{N} \leqslant \frac{(1-q)^w}{2}$, which means $N \geqslant \frac{2N_1}{(1-q)^w}$.

Then, $E[\bar{Z}] - \frac{N_1}{N} \geqslant \frac{(1-q)^w}{2}$. $P(N\bar{Z} < N_1) \leqslant e^{\frac{-N(1-q)^{2w}}{2}}$

We want $e^{\frac{-N(1-q)^{2w}}{2}}$ to be smaller than $\frac{1}{2n^{1+d}}$.

$$\frac{-N(1-q)^{2w}}{2} \leqslant ln(\frac{1}{2n^{1+d}})$$
$$N \geqslant -2\frac{ln(\frac{1}{2n^{1+d}})}{-2t^2}$$
$$= \frac{ln(\frac{1}{2}) - ln(n)(1+d)}{(1-q)^{2w}}$$

3.3 Unknown Variables in S

Before implementing the algorithm, we have to know the value of S. which is $\sum_{r=i-v}^{i-v+0.1w} P(R = r|F) \sum_{l=r+1}^{i-1} P(l-r,j)x_l$ from above. We can calculate $P(R = r|F)$ by using loops, and computers will do the rest of the job. F denotes the event that Y contains the substring.

$$P(R = r|F) = \frac{P(\{R = r\} \cap F)}{P(F)} \tag{6}$$

And we can see that $P(F)$ is:

$$P(F) = \sum_{k=w}^{n} P(F_k)$$

where F_k denotes the event that Y_j contains the substring at $y_{j-w-1}...y_j$. We then use loops to calculate the probability of each F_k. With the same logic,

$$P(R = r \cap F) = \sum_{k=w}^{n} P(F_k \cap \{R = r\})$$

But this time we already know that y_k has to come from x_r. Due to event E, $i - v \leqslant R \leqslant i - v + 0.1w$, in particular, $r < i$, thus we know that all entries of the substring are coming from entries of X that we know.

To solve the problem of F_k not being mutually exclusive, we apply the fact that X is w-substring unique. Denote event G as {for any a, no more than 0.1w entries are deleted in $X(a : a + w)$}. On G we know $i - v \leqslant R \leqslant i - v + 0.1w$, and therefore by lemma 3.3, the substring can only appear once in Y.

Instead of calculating $P(F_k)$, the algorithm computes $P(F_k \cap E)$.

$$P(F \cap E) = \sum_k P(F_k \cap E) = \sum_{r=i-v}^{i-v+0.1w} P(F_k \cap \{R = r\})$$

$$P(F \cap E) \leqslant P(F) \leqslant P(F \cap E) + \gamma$$

By (6),

$$0 \leqslant \frac{1}{P(F \cap E)} - \frac{1}{P(F)} = \frac{P(F) - P(F \cap E)}{P(F)P(F \cap E)} \leqslant \frac{\gamma}{P(F)P(F \cap E)}$$

$$S = \frac{1}{P(F)} \cdot \sum_{r=i-v}^{i-v+0.1w} P(\{R = r\} \cap F) \sum_{l=r+1}^{i-1} P(l-r, j) x_l$$

Error ϵ made when computing S:

$$\begin{aligned}
0 \leqslant \epsilon \leqslant & \frac{\gamma}{P(F)P(F \cap E)} \sum_{r=i-v}^{i-v+0.1w} P(\{R = r\} \cap F) \sum_{l=r+1}^{i-1} P(l-r, j) x_l \\
= & \frac{\gamma}{P(F \cap E)} \sum_{r=i-v}^{i-v+0.1w} \frac{P(\{R = r\} \cap F)}{P(F)} \sum_{l=r+1}^{i-1} P(l-r, j) x_l \\
= & \frac{\gamma}{P(F \cap E)} \sum_{r=i-v}^{i-v+0.1w} P(R = r | F) \sum_{l=r+1}^{i-1} P(l-r, j) x_l \\
= & \frac{\gamma}{P(F \cap E)} S
\end{aligned}$$

Since $P(F \cap E) \geqslant P(F) - \gamma \geqslant (1-q)^w - \gamma$ and $\gamma \leqslant \frac{(1-q)^w}{2}$,

$$\begin{aligned}
0 \leqslant \epsilon \leqslant & \frac{\gamma}{(1-q)^w - \gamma} \\
\leqslant & \frac{2\gamma}{(1-q)^w} \\
\leqslant & \frac{size of the gap}{4}
\end{aligned}$$

Because we already know that the smallest possible gap is $(\frac{1}{2})^{4w+3}$

$$\gamma \leqslant \frac{(1-q)^w}{2} \cdot \frac{(\frac{1}{2})^{4w+3}}{4} = \frac{1}{64} \cdot (\frac{1-q}{16})^w$$

How many steps are needed for getting $P(F \cap E)$?

Let k' be $0.2w$. Using Sterling's formula, we can get approximately number of steps are required.

$$
\begin{aligned}
\binom{1.2w}{w} &= \binom{k'+w}{k'} = \frac{(k'+w)!}{k'!w!} \\
&\leqslant \frac{e}{\sqrt{2\pi}} \cdot \frac{\sqrt{2\pi(k'+w)} \cdot (\frac{k'+w}{e})^{k'+w}}{\sqrt{2\pi(k')}(\frac{k'}{e})^{k'} \cdot \sqrt{2\pi(w)}(\frac{w}{e})^{w}} \\
&= \frac{e}{\sqrt{2\pi}} \cdot \frac{(k'+w)^{k'+w} \cdot \sqrt{k'+w} \cdot (\frac{1}{e})^{k'+w}}{\sqrt{2\pi k' w} \cdot k'^{k} \cdot w^{w} \cdot (\frac{1}{e})^{k'+w}} \\
&= \frac{e}{\sqrt{2\pi}} \cdot \frac{(k'+w)^{k'+w}}{k'^{k} \cdot w^{w}} \cdot \frac{1}{\sqrt{2\pi}} \cdot \sqrt{\frac{k'+w}{k'w}} \\
&= \frac{e}{\sqrt{2\pi}} \cdot \frac{1}{\sqrt{2\pi}} \cdot \frac{(1.2w)^{w} \cdot (1.2w)^{0.2w}}{(0.2w)^{0.2w} \cdot w^{w}} \cdot \sqrt{\frac{1.2}{0.2w}} \\
&= \frac{e}{\sqrt{2\pi}} \cdot \frac{1}{\sqrt{2\pi}} \cdot \sqrt{\frac{6}{w}} \cdot (1.2)^{w} \cdot 6^{0.2w} \\
&= \frac{e}{2\pi} \cdot \frac{1}{\sqrt{w}} \cdot \sqrt{6} \cdot (1.2 \cdot 6^{0.2})^{w} \leqslant 1.8^{w}
\end{aligned}
$$

Therefore, we need at least 1.8^w steps to get $P(F \cap E)$.

4 Algorithm

Algorithm 1 Reconstructing first i bits of a known length string

Input: N Received Traces

Variables: a known length of string n; a known deletion probability q; every bit of a trace; the original string X, which will be attained by adding elements in an array.

for j from 1 to floor$(1-3q)\cdot n+3q$ **do**

  x $=0$ ▷ number of times 1 is appearing at Y_j

  for k from 1 to N **do**

    if ${Y_j}^k = 1$ **then**

      $x+=1$

  $R[j-1] = \frac{x}{N}$ ▷ $P(Y_j = 1) = R[j-1]$

for i from 1 to n **do**

  $j = floor(1-3q)\cdot i + 3q$

  $P(i,j) = \binom{i-1}{j-1}\cdot q^{i-j}\cdot(1-q)^j$

  $s = R[j-1] - \sum_{l=j}^{i-1} P(l,j)\cdot X[l-1]$

  if $s \geqslant \frac{3P(i,j)}{4}$ **then**

    $X[i-1] = 1$

  else

    $X[i-1] = 0$

return X

Algorithm 2 Function that creates matching sublists

x_r and x_u matches to the last and the first bit of the string respectively.
WaysMatch = 0 ▷ number of ways of choosing $w-2$ entries between position u and r leading to the correct match.
Def: subsets(a, b)
if $a = 0$ **then**
    return [[]] ▷ empty set
else
    Lists = []
    for maxVal from a to b **do**
        ListsBeginning = subsets(a-1, maxValue -1)
        for sublist in ListsBeginning **do**
            Lists.append(sublist.append(maxValue))
    return Lists
ListOfSubsets = subsets$(w-2, r-u-1)$
for every subset of length $w-2$ in ListOfSubsets **do** ▷ total num of subsets: $\binom{r-u-1}{w-2}$
    IsThereAMatch = True
    for l in subset **do**
        if **then**$x_{u+l} != x_{i-v-w+1+l}$
            IsThereAMatch = False
    if IsThereAMatch **then**
        WaysMatch += 1
$P(F_k \cap \{R=r\} \cap \{U=u\}) =$ WaysMatch $\binom{u-1}{k-w}(1-q)^k q^{r-k}$ ▷ The first three terms represent the probability of keeping $k-w$ bits before x_u ▷ The last two terms represent the probability that bits in between x_u and x_r for any correct match.

Algorithm 3 Reconstructing X by using substring

Variables: a known length of string n; a known deletion probability q; every bit of a trace; the original string X from 1 to $i-1$; a known substring with length w. Known v.
$P(F_k \cap \{R=r\}) = \sum_u P(F_k \cap \{R=r\} \cap \{U=u\})$
$P(F \cap \{R=r\}) = \sum_k P(F_k \cap \{R=r\})$
$P(R=r|F) = \frac{1}{P(F)} P(F \cap \{R=r\})$
$S = \sum_{r=i-v}^{i-v+0.1w} P(R=r|F) \sum_{l=r+1}^{i-1} P(l-r,j) x_l$
if $P(Y_j^{new}=1) \leqslant \frac{\gamma}{2} + S + \frac{3}{4}\sum_r P(R=r|F)P(i-r,j)$ **then** ▷ threshold
    $X[i-1] = 0$
if $P(Y_j^{new}=1) \geqslant \frac{\gamma}{2} + S + \frac{3}{4}\sum_r P(R=r|F)P(i-r,j)$ **then**
    $X[i-1] = 1$