Frequency Moments - Approximations and Space Complexity

Felix Biermeier

Abstract The frequency moments characterize a sequence of elements by a number $F_k = \sum_{i=1}^{n} m_i^k$ where $m_i$ denotes the amount of each type of element $i$. We present the work of N. Alon, Y. Matias and M. Szegedy who construct randomized algorithms to approximate $F_k$ and analyze its space complexity [2]. It turns out that there exists approximation algorithms for certain frequency moments in logarithmic space. In particular for those frequency moments which are useful in data base applications. Unfortunately, the lower bound to approximate $F_k$ for $k > 5$ is at least $\Omega(n^{1-1/k})$.

1 Introduction

In this work we present estimates of the frequency moments of data streams by Noga Alon, Yossi Matias and Mario Szegedy [2]. Due to huge data sets where we cannot store the whole input in our memory it is only natural to consider a data streaming model. The frequency moment is simply the sum over the number of occurrences of each data value in the stream to the power of a certain integer number which we denote by $F_k = \sum_{i=1}^{n} m_i^k$ where $m_i$ counts the occurrence of each data values $i \in \{1, \ldots, n\}$. A special case is $F_\infty = \max_{1 \leq i \leq n} m_i$.

In general the frequency moment supplies statistical informations about the given data set. Due to the definition $F_0$ and $F_1$ are very intuitive where $F_0$ is the number of distinct elements and $F_1$ is the length of the whole data stream. The other frequency moments are less intuitive but not less interesting. These are indicators for the skew of data which is especially relevant in database applications. The frequency moment $F_2$ is used in [8] to estimate the size of query results.

The objective of this work is to present several approximation algorithms to estimate frequency moments. In particular we are interest in algorithms with sublinear space complexity due to the huge amount of data. The main focus lies on randomized algorithms. We will see that in comparison to deterministic algorithms the randomized algorithms are more efficient in regards to the space
This work is structured as follows. We start with approximations of the frequency moment $F_k$ in section 2 which contains additionally an improved approximation of $F_2$. After this we show lower bounds regarding the space complexity in section 3. In this section we rely most of the time on communication complexity. We also disclose the connection between randomized and deterministic algorithms mentioned before. The last section provides some additional remarks.

2 Approximations of frequency moments $F_k$

In this section we present randomized approximations to compute the frequency moment $F_k$. The objective is to construct those algorithms in sublinear space with respect to the length of the data stream and the number of different data values. A naive approach would be to consider a counter variable for every possible data value which is a simple approach but not space efficient unless we are only interested in $F_1$. By the definition of frequency moments $F_1$ denotes the length of the data stream. Therefore it is sufficient to use a single counter for all different values.

2.1 Approximating $F_k$

In this part we present and analyze an algorithm to approximate the frequency moments $F_k$. The general idea of this algorithm is to compute a good estimation of $F_k$ in expectation. Additionally the algorithm will repeat this procedure independently from each other and outputs the median of those results for each instance. This is a usual technique to boost the success probability of a certain event. It follows the whole statement.

**Theorem 1.** For every $k \geq 1$, every $\delta > 0$ and every $\varepsilon > 0$ there exists a randomized algorithm that computes, given a sequence $A = (a_1, \ldots, a_m)$ of members of $N = \{1, \ldots, n\}$ in one pass and using

$$O\left(\frac{k \log (1/\varepsilon)}{\delta^2} n^{1-1/k} \log n + \log m \right)$$

memory bits, a number $Y$ such that

$$\Pr[|Y - F_k| > \lambda F_k] \leq \varepsilon.$$ 

**Proof.** Let $s_1 = \frac{8kn^{1-1/k}}{\lambda^2}$ and $s_2 = 2 \log (1/\varepsilon)$. To simplify the proof we assume that the length of a sequence $A$ is known and we set $m = |A|$. We can get rid of this assumption but we omit the details. The algorithm to approximate $F_k$ works as follows.

**Algorithm 1 Estimate $F_k$**

```
1: for $i \leftarrow 1$ to $s_2$ do
2:     for $j \leftarrow 1$ to $s_1$ do
3:         $a_p \leftarrow \frac{\ell}{j} A$
4:         $r \leftarrow |\{q : q \geq p, a_q = a_p\}|$
5:         $X_{ij} \leftarrow m (r^k - (r - 1)^k)$
6:     end for
7:     $Y_i \leftarrow \frac{1}{s_1} \sum_{j=1}^{s_1} X_{ij}$
8: end for
9: output median of $(Y_1, \ldots, Y_{s_2})$
```
The inner loop of the algorithm computes a random variable $Y_i$ which consists of the average of independent and identically distributed random variables $X_{ij}$. The outer loop repeats this procedure for a certain amount of runs. Finally the algorithm outputs the median of those $Y_i$. We can describe this approach by the median of means. The use of the median is a usual technique to boost the success probability of a certain event which we will see in the analysis. The main issue of the algorithm is to track a random value $a_p$ from the sequence and count the occurrence of $a_p$ in the sequence $\mathcal{A}$ subsequently. In order to compute a certain $X_{ij}$ we maintain the current value $a_p \in \mathcal{N}$ and a counter $r \in \{1, \ldots, m\}$. This requires $\mathcal{O} (\log n + \log m)$ space.

The analysis of the algorithm is quite simple. We are interested in the expected value and variance of a single $Y_i$. This requires the analysis of $X_{ij}$ To simplify the notation we set $X = X_{ij}$. Fortunately, the expected value of $X$ supplies already a good result to estimate $F_k$

$$E[X] = \frac{1}{m} \sum_{i=1}^{n} \sum_{j=1}^{m} m(j^k - (j - 1)^k)$$

$$= \frac{m}{m}[(1^k + (2^k - 1^k)) \cdots + (m^k - (m_1 - 1))^k] +
(1^k + (2^k - 1^k)) \cdots + (m_2^k - (m_2 - 1)^k) + \cdots +
(1^k + (2^k - 1^k)) \cdots + (m_n^k - (m_n - 1)^k)]$$

$$= \sum_{i=1}^{n} m_i^k = F_k.$$

With regards to the variance we consider the definition $\text{Var}[X] = E[X^2] - E[X]^2$ where the expected value of $X^2$ is

$$E[X^2] = \frac{1}{m} \sum_{i=1}^{n} \sum_{j=1}^{m} (m(j^k - (j - 1)^k))^2$$

$$= \frac{m^2}{m}[(1^{2k} + (2^k - 1)^2) \cdots + (m_1^k - (m_1 - 1)^2)] +
(1^{2k} + (2^k - 1)^2) \cdots + (m_2^k - (m_2 - 1)^2) + \cdots +
(1^{2k} + (2^k - 1)^2) \cdots + (m_n^k - (m_n - 1)^2)].$$

We can estimate this term by using the following inequality

$$a^k - b^k \leq (a - b)k a^{k-1}, a > b > 0$$

which leads to

$$E[X^2] \leq m((k 1^{2k-1} + \cdots + km_1^{k-1}(m_1^k - (m_1 - 1)^k)) + \cdots +
(k 1^{2k-1} + \cdots + km_n^{k-1}(m_n^k - (m_n - 1)^k))]
= km[(1^{2k-1} + \cdots + m_1^{k-1}(m_1^k - (m_1 - 1)^k)) + \cdots +
(1^{2k-1} + \cdots + m_n^{k-1}(m_n^k - (m_n - 1)^k))]
\leq km[(m_1^{2k-1} + \cdots + m_1^{k-1}(m_1^k - (m_1 - 1)^k)) + \cdots +
(m_n^{2k-1} + \cdots + m_n^{k-1}(m_n^k - (m_n - 1)^k))]
= km \sum_{i=1}^{n} m_i^{k-1} m_i^k = km \sum_{i=1}^{n} m_i^{2k-1} = km F_{2k-1} = k F_1 F_{2k-1}.$$
\[ E[Y_i] = E \left[ \frac{1}{s_1} \sum_{j=1}^{s_1} X_j \right] = \frac{1}{s_1} \sum_{j=1}^{s_1} E[X_j] = F_k. \]

This is a crucial observation and it allows us the computation of \( \text{Var}[Y_i] \) with the fact below

\[
\left( \sum_{i=1}^{n} m_i \right) \left( \sum_{i=1}^{n} m_i^{2k-1} \right) \leq n^{1-1/k} \left( \sum_{i=1}^{n} m_i^k \right)^2. \tag{1}
\]

Therefore by the independence of the random variables \( X_j \) for \( Y_i \) we get

\[
\text{Var}[Y_i] = \text{Var} \left[ \frac{1}{s_1} \sum_{j=1}^{s_1} X_j \right] = \frac{1}{s_1^2} \sum_{j=1}^{s_1} \text{Var}[X_j] = \frac{1}{s_1^2} \sum_{j=1}^{s_1} E[X_j^2] - E[X_j]^2 \leq \frac{E[X_j^2]}{s_1} \leq \frac{kF_1F_{2k-1}}{s_1}
\]

\[
\leq \frac{kn^{1-1/k}F_k^2}{s_1}. \tag{1}
\]

Together with the observation above the Chebyshev’s Inequality supplies for every fixed \( i \) and \( s_1 = \frac{8kn^{1-1/k}}{\lambda^2} \)

\[
\text{Pr}[|Y_i - F_k| > \lambda F_k] \leq \frac{\text{Var}[Y_i]}{(\lambda F_k)^2} \leq \frac{kn^{1-1/k}F_k^2}{s_1\lambda^2 F_k^2} \leq \frac{1}{8}.
\]

Remember that the algorithm outputs the median of all \( Y_i \). The Chernoff Bound will show that the median is a good choice as an output in most of the cases. To apply Chernoff Bound we define by \( Z_i \) a random variable

\[ Z_i = 1 \iff |Y_i - F_k| > \lambda F_k \]

such that \( Z = \sum_{i=1}^{s_2} Z_i \). All \( Z_i \) are independent binary random variables since all \( Y_i \) respectively all \( X_{ij} \) are independent. With respect to our desired result \( Z_i \) describes a bad event where our estimation \( Y_i \) deviates too much from the frequency moment \( F_k \). The expectation of \( Z \) is

\[ E[Z] = \sum_{i=1}^{s_2} E[Z_i] \leq \frac{s_2}{8}. \]

By choosing \( \delta = 3 \) and \( \mu = E[Z] \) Chernoff bound supplies

\[
\text{Pr} \left[ Z \geq (1 + 3) \frac{s_2}{8} = \frac{s_2}{2} \right] \leq e^{-\frac{9s_2/8 - \mu}{\delta^2 \mu/2}} = e^{-\frac{27 \log(1/\varepsilon)}{8\mu}} = e^{-\frac{27 \log(1/\varepsilon)}{16 \mu/16}} \leq \varepsilon, 0 < \varepsilon < 1
\]

which is sufficient for all relevant \( \varepsilon \). Therefore if the number of bad events is less than \( s_2/2 \) the median \( Y \) supplies a good estimation for \( F_k \) with the probability

\[ \text{Pr} [|Y - F_k| \leq \lambda F_k] \geq 1 - \varepsilon. \]

\( \square \)
2.2 Improved space bound for $F_2$

While Theorem 1 already supplies a relatively good approximation with a sublinear space constraint it is possible to improve this result for certain $F_k$. In this section we present such a case for $F_2$. The algorithm basically uses a similar approach what we have seen before but the space complexity is decreased to a logarithmic term.

**Theorem 2.** For every $\delta > 0$ and every $\varepsilon > 0$ there exists a randomized algorithm that computes, given a sequence $A = (a_1, \ldots, a_m)$ of members of $N = \{1, \ldots, n\}$ in one pass and using

$$O\left(\frac{\log (1/\varepsilon)}{\lambda^2} \log n + \log m\right)$$

memory bits, a number $Y$ such that

$$\Pr[|Y - F_2| > \lambda F_2] \leq \varepsilon.$$

**Proof.** Similar to the previous proof we set the parameter $s_1$ and $s_2$ to $s_1 = \frac{16}{\lambda^2}$ respectively $s_2 = 2\log (1/\varepsilon)$. The algorithm to approximate $F_2$ works as follows.

**Algorithm 2 Estimate $F_2$**

1: for $i \leftarrow 1$ to $s_2$ do
2: for $j \leftarrow 1$ to $s_1$ do
3: $v_p = (\varepsilon_1, \ldots, \varepsilon_n) \overset{R}{\rightarrow} V$
4: $Z \leftarrow \left(\sum_{l=1}^{n} \varepsilon_l m_l\right)^2$
5: $X_{ij} \leftarrow Z^2$
6: end for
7: $Y_i \leftarrow \frac{1}{s_1} \sum_{j=1}^{s_1} X_{ij}$
8: end for
9: output median of $(Y_1, \ldots, Y_{s_2})$

Similar to the previous algorithm in section 2.1 $Y_i$ and $X_{ij}$ are random variables while $X_{ij}$ are independent and identically distributed. The main difference between those two algorithms is the computation of $X_{ij}$. The set $V$ of size $h \in O(n^2)$ consists of vectors $v_i$ where each entry is a random variable $\varepsilon_l \in \{-1, 1\}$. Additionally the random variables $\varepsilon_l$ are four-wise independent. This will be sufficient to prove the statement. The construction of such a set is based on arithmetic operations in a finite field which is rather technical and can be find in [1]. In order to fulfill the space constraint the random variables $X_{ij}$ are computed in two steps. Since $Z$ is a linear function it is only necessary to maintain the current sum of $Z$ and the current value $p$ to get the relative $\varepsilon_l$. This is possible in $O(\log n + \log m)$ space.

As in the proof before we compute the expected value and variance of $Y_i$. Due to the definition of $Y_i$ the expected value is

$$E[Y_i] = E\left[\frac{1}{s_1} \sum_{j=1}^{s_1} X_{ij}\right] = \frac{1}{s_1} \sum_{j=1}^{s_1} E[X_{ij}] = E[X_j]$$

where the expected value of a single $X_{ij}$ is
This leads to

$$E[X] = E \left[ \left( \sum_{i=1}^{n} \varepsilon_i m_i \right)^2 \right] = \sum_{i=1}^{n} m_i^2 E[\varepsilon_i^2] + 2 \sum_{1 \leq i < j \leq n} m_i m_j E[\varepsilon_i \varepsilon_j] = \sum_{i=1}^{n} m_i^2 = F_2.$$  

By the definition of the variance we have to compute the term

$$E[X^2] = E \left[ \left( \sum_{i=1}^{n} \varepsilon_i m_i \right)^4 \right] = E \left[ \left( \sum_{i=1}^{n} \varepsilon_i m_i \right) \left( \sum_{j=1}^{n} \varepsilon_j m_j \right) \left( \sum_{k=1}^{n} \varepsilon_k m_k \right) \left( \sum_{l=1}^{n} \varepsilon_l m_l \right) \right]$$

$$= \sum_{i=1}^{n} m_i^4 E[\varepsilon_i^4] + \sum_{1 \leq i < j \leq n} 4 m_i^4 m_j^4 E[\varepsilon_i \varepsilon_j^2] + \sum_{1 \leq i < j \leq n} 6 m_i^2 m_j^2 m_k^2 E[\varepsilon_i^2 \varepsilon_j^2 \varepsilon_k^2] + \sum_{1 \leq i < j < k \leq n} 24 m_i m_j m_k m_l E[\varepsilon_i \varepsilon_j \varepsilon_k \varepsilon_l]$$

$$= \sum_{i=1}^{n} m_i^4 + 6 \sum_{1 \leq i < j \leq n} m_i^2 m_j^2.$$  

This leads to

$$\text{Var}[X] = E[X^2] - E[X]^2 = \sum_{i=1}^{n} m_i^4 + 6 \sum_{1 \leq i < j \leq n} m_i^2 m_j^2 - \left( \sum_{i=1}^{n} m_i^2 \right)^2$$

$$= \sum_{i=1}^{n} m_i^4 + 6 \sum_{1 \leq i < j \leq n} m_i^2 m_j^2 - \sum_{i=1}^{n} m_i^4 - 2 \sum_{1 \leq i < j \leq n} m_i^2 m_j^2$$

$$= 4 \sum_{1 \leq i < j \leq n} m_i^2 m_j^2 + 2 \sum_{i=1}^{n} m_i^4 = 2 \left( \sum_{1 \leq i < j \leq n} m_i^2 m_j^2 + \sum_{i=1}^{n} m_i^4 \right)$$

$$= 2 \left( \sum_{i=1}^{n} m_i^2 \right)^2 = 2F_2^2.$$  

By using the observation from above we obtain

$$\text{Var}[Y] = \frac{1}{s_1} \sum_{i=1}^{n_1} X_i = \frac{1}{s_2} \sum_{i=1}^{n_2} \text{Var}[X_i] = \frac{1}{s_1} \text{Var}[X_i] = \frac{1}{s_1} \text{Var}[X_i] \leq \frac{2F_2^2}{s_1}. $$

This is sufficient to apply the Chebyshev Inequality such that

$$\Pr[|Y - F_2| > \lambda F_2] \leq \frac{\text{Var}[Y_i]}{(\lambda F_2)^2} \leq \frac{2F_2^2}{s_1 \lambda^2 F_2^2} = \frac{1}{8}.$$  

Similar to the previous proof the Chernoff Bound completes the proof.

### 2.3 Approximating \( F_0 \)

In this section we present the missing frequency moment \( F_0 \) which is not covered by the algorithms above. Due to its definition of \( F_0 \) we cannot apply the approach from before. However, there exists other approaches and the space complexity is not less exciting since we can construct an algorithm
with logarithmic space constraint. The crucial part of the algorithm is the existence of a family of linear hash functions with certain properties.

**Theorem 3.** For every \( c > 2 \) there exists a randomized algorithm that, given a sequence \( A \) of members \( N \), computes a number \( Y \) using \( O(\log n) \) memory bits, such that the probability that the ratio between \( Y \) and \( F_0 \) is not between \( 1/c \) and \( c \) is at most \( 2c \).

**Proof.** Consider a finite field \( \mathcal{F} = \mathbb{GF}(2^d) \) with the smallest \( d \in \mathbb{N} \) such that \( 2^d > n \) and \( N = \{1, \ldots, n\} \) is a subset of \( \mathcal{F} \). This allows computations of arithmetic operations in \( \mathcal{F} \) with members of the sequence \( A \). This can be observed in the algorithm below.

**Algorithm 3 Estimate \( F_0 \)**

1: \( a, b \xleftarrow{\text{R}} \mathcal{F} \) \hspace{1cm} \( \triangleright \) independent
2: \( R \leftarrow 0 \)
3: for \( i \leftarrow 1 \) to \( m \) do
4: \( z_i \leftarrow a \cdot a_i + b \)
5: \( r_i \leftarrow r(z_i) \) \hspace{1cm} \( \triangleright r(z) \) denotes the largest number of rightmost bits which are all 0
6: if \( r_i > R \) then
7: \( R \leftarrow r_i \)
8: end if
9: end for
10: output \( 2R \)

We choose \( a, b \in \mathcal{F} \) uniformly at random and independently. The general idea of this algorithm is the computation of linear hash functions \( z_i \) over the finite field \( \mathcal{F} \) and members of the sequence \( A \). This is a random mapping of elements \( a_i \in \mathcal{A} \) to \( z_i \) and it possesses some useful properties which will be observed later.

An other interesting point is the function \( r(z) \) which denotes the largest number of rightmost bits of a binary vector where those bits are all 0. This function affects the output of the algorithm. Regarding the space complexity it is sufficient to maintain the values \( a, b \) and the current binary index of \( R \). This is possible in \( O(\log n) \) respectively \( O(\log \log n) \) space.

For the rest of the proof we assume \( F_0 \) is the correct value of distinct elements in \( \mathcal{A} \) and \( r \) is fixed. The analysis is based on two properties of \( z_i \) which were indicated before. The random mapping from \( a_i \) to \( z_i \) is uniformly distributed over \( \mathcal{F} \) if both values are fixed. This implies for \( r \in \mathbb{N} \)

\[
\Pr [r(z_i) \geq r] = \frac{1}{2^r}.
\]

The other property is the pairwise independence for two distinct and fixed elements \( a_i \) and \( a_j \) which implies

\[
\Pr [r(z_i) \geq r \land r(z_j) \geq r] = \frac{1}{2^{2r}}.
\]

In order to continue it is necessary to define a random variable with regards to the number of distinct elements \( F_0 \). Let \( W_{a_i} \) be a random variable

\[
W_{a_i} = 1 \iff r(z_i) \geq r
\]

such that \( Z_r = \sum W_{a_i} \) where each value of all \( F_0 \) distinct elements is count only once. This leads to

\[
E[Z_r] = \sum E[W_{a_i}] = \frac{F_0}{2^r}.
\]
Due to the pairwise independence of $z_i$ and $z_j$ the variance of $Z_r$ is

$$\text{Var}[Z_r] = \text{Var} \left[ \sum W_{a_i} \right] = \sum \text{Var} [W_{a_i}] = \sum \frac{1}{2^r} \left( 1 - \frac{1}{2^r} \right) = \frac{F_0}{2^r} \left( 1 - \frac{1}{2^r} \right) \leq \frac{F_0}{2^r}.$$ 

To archive the desired result it is necessary to bound the probability of

$$\frac{2^r}{F_0} \notin \left[ \frac{1}{c}, c \right].$$

We distinguish between the probabilities of $Z_r = 0$ and $Z_r > 0$. The inequality $c2^r < F_0$ and the Chebyshev inequality implies

$$\Pr[Z_r = 0] \leq \Pr \left[ |Z_r - E[Z_r]| \geq E[Z_r] \right] \leq \frac{\text{Var}[Z_r]}{E[Z_r]^2} < \frac{1}{E[Z_r]} = \frac{2^r}{F_0} < \frac{1}{c},$$

while the inequality $2^r > cF_0$ and the Markov inequality implies

$$\Pr[Z_r > 0] = \Pr[Z_r \geq 1] \leq \frac{E[Z_r]}{1} = \frac{F_0}{2^r} < \frac{1}{c}.$$ 

By using the union bound we archive

$$\Pr \left[ \frac{2^r}{F_0} \notin \left[ \frac{1}{c}, c \right] \right] < \frac{1}{c} + \frac{1}{c} = \frac{2}{c}.$$ 

In case the algorithm outputs $Y = 2^R$ for which $Z_r > 0$ we obtain our desired result. □

3 Lower bounds

In the previous section we have seen a couple of randomized approximations to compute the frequency moment $F_k$ in sublinear space. The question arises if we can establish lower bounds for $F_k$. We show a rather simple case for $F_\infty$ and we continue with the general cases of $F_k$. Especially the latter cases require a deeper introduction to communication complexity. But before we start with a statement which demonstrates the benefits of randomization in the approximations above.

3.1 Space complexity of deterministic algorithms

In this section we present the necessity of randomization in all algorithms of section 2. If we use a deterministic algorithm it is impossible for almost all frequency moments $F_k$ to archive a sublinear space constraint. The only exception is $F_1$ what is the length of the data stream and it can compute in logarithmic space by using a single counter.

**Proposition 1.** For any nonnegative integer $k \neq 1$, any deterministic algorithm that outputs, given a sequence $A$ of $n/2$ elements of $N = \{1, \ldots, n\}$, a number $Y$ such that $|Y - F_k| \leq 0.1F_k$ must use $\Omega(n)$ memory bits.

**Proof.** By results of coding theory there exists a family $\mathcal{G}$ of $t = 2^{\Omega(n)}$ subsets of $N$ such that each $G \in \mathcal{G}$ has a cardinality of at most $n/4$ and for all $G_1 \neq G_2$ holds

$$|G_1 \cap G_2| \leq n/8.$$
We fix a deterministic algorithm $D$ from above and define a input sequence $A(G_1, G_2)$ with length $n/2$ for all two members of $G$. The length of such an input is by definition $n/2$. We will compare the memory configurations of two distinct input sequences after the algorithm finished the first $n/4$ elements respectively the first subset $G$ of $A$ and the whole input.

We assume that the size of the memory is at most $\log t$. Imagine that the algorithm $D$ produces for each $G \in G$ a memory configuration and we can give each configuration a number. Since the size of $G$ is $t$ the assumption together with the pigeonhole principle imply that there exists at least two distinct sets $G_1, G_2 \in G$ for which the algorithm $D$ produces the same memory configuration.

We consider two input sequences $A(G_1, G_1)$ and $A(G_2, G_1)$ where $G_1, G_2 \in G$ are distinct. By the argument above the algorithm produces for the first half of both inputs the same memory configuration and eventually the same result for the whole input. The first input $A(G_1, G_1)$ supplies

\[ F_0 = n/4, \]
\[ F_k = \sum_{i \in G_1} 2^k = 2^k \cdot |G_1| = 2^k \cdot n/4 \]

while the second input $A(G_2, G_1)$ supplies

\[ F_0 \geq \frac{n}{2} - \frac{n}{8} = \frac{3n}{8}, \]
\[ F_k \leq \left( \frac{n}{2} - \frac{2n}{8} \right) + 2^k \cdot \frac{n}{8} = \frac{n}{4} + 2^k \cdot \frac{n}{8}. \]

We denote by $F_k^{(1)}$ and $F_k^{(2)}$ for $k \in \mathbb{N}$ the output of the algorithm $D$ under the first respectively second input sequence. By comparing both outputs we can see that they are unequal

\[ F_0^{(1)} = \frac{n}{4} < \frac{3n}{8} \leq F_0^{(2)} \]
\[ F_k^{(1)} = 2^k \cdot \frac{n}{4} > 2^k \cdot \left( \frac{n}{8} + \frac{n}{4 \cdot 2^k} \right) = 2^k \cdot \left( \frac{n}{8} + \frac{n}{4 \cdot 2^k} \right) = \frac{n}{4} + 2^k \cdot \frac{n}{8} \geq F_k^{(2)}, k > 1. \]

This is a contradiction to the assumption that the algorithm $D$ outputs the same value for both inputs and it follows that the algorithm must use at least $\log t = \Omega(n)$ memory bits. \qed

### 3.2 Space complexity of $F_\infty$

In this section we give a brief introduction to communication complexity which we can apply to the special case $F_\infty$. Just for a reminder $F_\infty$ describes the maximum number of occurrences of an item value over all possible item values in the data stream. The definitions below will help us to set up a foundation for the upcoming statements. About that the first definition is in general useful to analyze communication complexity of certain problems. It is based on [7].

**Definition 1 (probabilistic communication complexity).** We have given a boolean function $f : \{0, 1\}^n \times \{0, 1\}^n \to \{0, 1\}$, two players with unlimited computation power and inputs $x, y \in \{0, 1\}^n$. Both players know their own input $x$ respectively $y$ and they are allowed to send messages to each other according to a probabilistic protocol. At the end they output the value of $f(x, y)$. With a probability of at least $1 - \varepsilon$ this output is the correct value. We denote by $C_\varepsilon(f)$ the $\varepsilon$-error probabilistic communication complexity of $f$ which is defined as the expected number of bits communicated in the worst-case.
Definition 2 (Disjointness problem). We have given the disjointness function \( DIS_n : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\} \) and two inputs \( x, y \in \{0,1\}^n \). Both inputs characterize a subset \( N_x \) respectively \( N_y \) of \( \{1, \ldots, n\} \). The function \( DIS_n \) outputs 1 iff \( N_x \) and \( N_y \) intersect.

Together with a result of [4] for the disjointness problem it follows a lower bound for \( F_\infty \).

**Proposition 2.** Any randomized algorithm that outputs, given a sequence \( A \) of at most \( 2n \) elements of \( N = \{1, \ldots, n\} \) a number \( Y \) such that

\[
\Pr[|Y - F_\infty| \leq F_\infty/3] > 1 - \varepsilon
\]

for some fixed \( \varepsilon < 1/2 \) must use \( \Omega(n) \) memory bits.

**Proof.** We have given an algorithm \( D \) as above and two players with binary inputs \( x \) respectively \( y \) which characterize subsets \( N_x, N_y \subseteq \{1, \ldots, n\} \). We will reduce the disjointness problem to the \( F_\infty \) problem by defining a communication protocol between those two players. Let \( A \) be a sequence of length at most \( 2n \) consisting of all elements in \( N_x \) and \( N_y \) without mixing or ordering both subsets. The communication protocol is quite simple. It starts with the first player who runs the algorithm \( D \) on the first \( |N_x| \) elements of \( A \). After this it sends its content of memory to the second player who applies the algorithm \( D \) on the rest of \( A \) and outputs \( DIS_n(x, y) \). According to the disjointness function we distinguish between two results

\[
DIS_n(x, y) = \begin{cases} 
0 & \text{if } D \text{ outputs a value } < 4/3 \\
1 & \text{else,}
\end{cases}
\]

The first case implies that both sets \( N_x \) and \( N_y \) are disjoint which means that the correct value of \( F_\infty \) is 1 and \( |Y - F_\infty| \leq F_\infty/3 \) holds for the required probability.

The second case implies that both sets \( N_x \) and \( N_y \) intersect which means that the correct value of \( F_\infty \) is 2 and \( |Y - F_\infty| \leq F_\infty/3 \) holds for the required probability as well.

This shows that the \( F_\infty \) problem is at least as hard as the disjointness problem. Regarding the result from [4] we know that the disjointness problem needs at least \( \Omega(n) \) memory bits. Therefore we get a lower bound for \( F_\infty \) by the reduction above and we obtain our desired result. \( \square \)

### 3.3 Space complexity of \( F_k \)

In this section we analyze the space complexity of the general frequency moments \( F_k \). To do this we have to do more preliminary. First we introduce the distributional communication complexity which gives us a lower bound for the probabilistic communication complexity \( C_\varepsilon(f) \). The correctness behind this is based on Yao’s Minimax Principle which gives us the possibility to establish lower bounds for randomized algorithm by analyzing the performance of deterministic algorithms. The arguments are based on [3, 5, 6].

**Definition 3 (distributional communication complexity).** Similar to Definition 1 we have given a boolean function \( f \), two players and inputs \( x, y \in \{0,1\}^n \). But in this case both players use a deterministic protocol to communicate with each other. At the end they output the correct value of \( f(x, y) \) for all input pairs \( (x, y) \) except for at most an \( \varepsilon \)-fraction of the input under a probability distribution \( \mu \). We denote by \( D_\varepsilon(f|\mu) \) the \( \varepsilon \)-error distributional communication complexity for \( f \) under a distribution \( \mu \).

It will be necessary to change the communication game in comparison to the one we used in Proposition 2. The core of this game treats still a disjointness problem but we adapt the number of players and the objective itself.
**Definition 4 (communication game).** We have given \(s,t,n \in \mathbb{N}\) and \(N = \{1, \ldots, n\}\). We call \(DIS(s,t)\) a communication game with \(s\) players \(P_1, \ldots, P_s\). Each player has an input \(A_i\) with \(A_i \subseteq N\) and \(|A_i| = t\). The goal is to distinguish between disjoint respectively uniquely intersecting input sequences \((A_1, \ldots, A_s)\). The players are allowed to communicate with each other according to a probabilistic protocol. At the end of the protocol player \(P_s\) outputs a value.

Before we finally proof the main result of this section we need a last additional definition which helps us to quantify the quality of a probabilistic protocol in regards to the communication game \(DIS\).

**Definition 5 (\(\varepsilon\)-correct protocol).** We have given the communication game \(DIS(s,t)\). A probabilistic protocol is \(\varepsilon\)-correct if the protocol outputs 0 for any disjoint input respectively it outputs 1 for any uniquely intersecting input with probability at least \(1 - \varepsilon\). The output of other inputs may be arbitrary.

**Theorem 4.** For any fixed \(k > 5\) and \(\delta < 1/2\), any randomized algorithm that outputs, given an input sequence \(A\) of at most \(n\) elements of \(N = \{1, \ldots, n\}\) a number \(Z_k\) such that

\[
\Pr[|Z_k - F_k| > 0.1F_k] < \delta
\]

uses at least \(\Omega(n^{1-5/k})\) memory bits.

**Proof.** Given an algorithm \(\mathcal{D}\) as above we define a randomized protocol for \(DIS(s,t)\) where \(\mathcal{D}\) uses \(M\) memory bits, \(n = (2t-1)s + 1\) with \(s = n^{1/k}, t \in \Theta(n^{1-1/k})\) and \(n\) is sufficiently large enough. To simplify the proof we assume \(t = n^{1-1/k}\).

By the definition of \(DIS(s,t)\) we have \(s\) players \(P_1, \ldots, P_s\) with corresponding inputs \(A_1, \ldots, A_s\) where each \(A_i\) is a subset of \(N\) with a cardinality of \(t\). The communication between the players is defined by the following protocol. The first player \(P_1\) runs the algorithm \(\mathcal{D}\) on his input \(A_1\) and he sends his content of memory to the second player \(A_2\). This continues until the last player \(P_s\) outputs the final value \(Z_k\) of \(\mathcal{D}\).

According to \(DIS(s,t)\) we distinguish between two results

\[
DIS(s,t) = \begin{cases} 0 & \text{if } Z_k \leq 1.1st \\ 1 & \text{else} \end{cases}.
\]

The first case implies that the inputs \(A_1, \ldots, A_s\) are disjoint which means that the correct value of \(F_k\) is \(st\) and \(|Z_k - F_k| > 0.1F_k\) holds for the required probability. The second case implies that the inputs \(A_1, \ldots, A_s\) are uniquely intersecting which means that the correct value of \(F_k\) is

\[
F_k = s^k + s(t-1) = (2t-1)s + 1 + s(t-1) = s(3t-2) + 1 > s(3t-2) \geq \left(\frac{3}{2} + a\right)n, a \in o(1)
\]

and \(|Z_k - F_k| > 0.1F_k\) holds for the required probability as well.

Therefore the algorithm \(\mathcal{D}\) approximates \(F_k\) with probability at least \(1 - \gamma\) which implies that the protocol for \(DIS(s,t)\) is \(\gamma\)-correct. With regards to the communication complexity the protocol uses at most \(sM\) memory bits. By applying Proposition 3 which we will proof later we get a lower bound \(\Omega(t/s^2)\) for the amount of memory bits. This leads to the desired result

\[
M \geq \Omega(t/s^2) = \Omega(n/s^5) = \Omega(n^{1-5/k}).
\]

\(\square\)

The rest of this section contains all statements which are necessary to complete the proof above and still unproved.
Proposition 3. For any fixed $\varepsilon < 1/2$ and any $t \geq s^3$, the length of any randomized $\varepsilon$-correct protocol for the communication problem $\text{DIS}(s,t)$ is at least $\Omega(t/s^3)$.

Proof. We define a probability distribution $\mu$ on the input sequence $(A_1, \ldots, A_s)$ according to $\text{DIS}(s,t)$. We consider partitions $P$ of $N = \{1, \ldots, n\}$ such that $P = \bigcup_{j=1}^{s} I_j \cup \{x\}$ where each $I_j$ has cardinality of $2t-1$ and all $I_j$ and $\{x\}$ are pairwise disjoint. We choose such a partition $P$ uniformly at random of all those partitions of $N$. Furthermore we choose a subset $\overline{A}_j$ of cardinality $t$ of $I_j$ uniformly at random. At last we define for both cases with probability $1/2$

$$A_j = \begin{cases} \overline{A}_j & \forall j : 1 \leq j \leq s \\ (I_j - \overline{A}_j) \cup \{x\} & \forall j : 1 \leq j \leq s. \end{cases}$$

This shows that the distribution $\mu$ generates disjoint respectively uniquely intersecting input sequences $(A_1, \ldots, A_s)$ with probability $1/2$ in both cases. An other aspect is that each of the disjoint respectively uniquely intersecting input has the same probability. To distinguish between both inputs we denote by $(A_1^0, \ldots, A_s^0)$ disjoint input sequences and by $(A_1^1, \ldots, A_s^1)$ uniquely intersecting input sequences. Furthermore we define a box as a family $\overline{X}_1 \times \cdots \times \overline{X}_s$ where each $\overline{X}_i$ is a set of subsets of cardinality $t$ of $N$.

It is possible to show that all input sequences with a fixed and corresponding communication between players according to $\text{DIS}(s,t)$ forms a box. Regarding the distributional communication complexity problem it suffices to show that every deterministic protocol with less than $\Omega(t/s^3)$ communication bits errs with probability $\Omega(1)$ where we apply inputs according to the distribution $\mu$.

As we have mentioned above each box contains a fixed communication pattern. We are looking for communication patterns where the relative protocol outputs 0 on an input sequence $(A_1^0, \ldots, A_s^0)$ which means that the protocol errs. If the number of patterns which output 0 is less than $\frac{p}{s} 2^{ct/s^3}$ we conclude by summing up Lemma 1 over all boxes that

$$\Pr[\text{output 0 on input } (A_1^0, \ldots, A_s^0)] \geq \frac{1}{2e} \Pr[\text{output 0 on input } (A_1^1, \ldots, A_s^1)] - p.$$ 

By choosing a sufficiently small constant $p > 0$ we have shown the desired result. \qed

Lemma 1. There exists an absolute constant $c > 0$ such that for every box $\overline{X}_1 \times \cdots \times \overline{X}_s$

$$\Pr[(A_1^1, \ldots, A_s^1) \in \overline{X}_1 \times \cdots \times \overline{X}_s] \geq \frac{1}{2e} \Pr[(A_1^0, \ldots, A_s^0) \in \overline{X}_1 \times \cdots \times \overline{X}_s] - s2^{-ct/s^3}.$$ 

Proof. We fix a box $\overline{X}_1 \times \cdots \times \overline{X}_s$. A partition $P$ is $j$-bad for a constant $c > 0$ if

$$\Pr_P[A_j^1 \in \overline{X}_j] < \left(1 - \frac{1}{s+1}\right) \Pr_P[A_j^0 \in \overline{X}_j] - s2^{-ct/s^3}, \forall j, 1 \leq j \leq s$$

where $Pr_P(\cdot)$ denotes the conditional probability of a certain event given a partition $P$. Similar to this a partition $P$ is bad if the partition $P$ is $j$-bad for at least one $j$ respectively a partition is good if it is not $j$-bad for all $j$. According to these definitions we define random variables $\chi(P)$ and $\chi_j(P)$ such that

$$\chi(P) = 1 \iff P \text{ is bad partition},$$

$$\chi_j(P) = 1 \iff P \text{ is } j \text{-bad}.$$ 

It is obvious that $\chi(P) \leq \sum_{j=1}^{s} \chi_j(P)$. To simplify the notation we denote two events by
\[ \mathcal{E}' = (A'_1 \times \cdots \times A'_s) \in \overline{X_1} \times \cdots \times \overline{X_s}, \]
\[ \mathcal{E} = (A_1' \times \cdots \times A'_s) \in \overline{X_1} \times \cdots \times \overline{X_s}. \]

The crucial observation is that the partition \( P \) is chosen uniformly at random which leads to
\[
\Pr[\mathcal{E}] = \sum_P \Pr_P[\mathcal{E}] \cdot \Pr[\mathcal{P}] = \frac{1}{\#P} \sum_P \Pr_P[\mathcal{E}] \overset{(\text{Def})}{=} \mathbb{E}[\Pr_P[\mathcal{E}]].
\]

By Lemma 3 and the equality 2 above it follows that
\[
\Pr[\mathcal{E}] \geq \mathbb{E}[\Pr_P[\mathcal{E}](1 - \chi(P))] \geq \frac{1}{c} \mathbb{E}[\Pr_P[\mathcal{E}'](1 - \chi(P))] - s2^{-ct/s^3}. \tag{3}
\]

Keeping this in mind we continue with the term \( \mathbb{E}[\Pr_P[\mathcal{E}']\chi_j(P)] \) and we consider Lemma 2 to bound \( \chi_j(P) \). Given the information on the partition \( P \) it is enough to analyze the conditional probability \( \Pr_P[A'_0 \in \overline{X}_j] \). Due to Lemma 2 the choice of \( x \) of the union \( I_j \cup \{x\} \) is still left. We denote by \( l \) the number of subsets of cardinality \( l \) in \( \overline{X}_j \) which are part of the union \( I_j \cup \{x\} \). This implies
\[
\Pr_P[A'_0 \in \overline{X}_j] = l/(2t) .
\]

Even any choice of \( x \) changes not much but only that
\[
\Pr_P[A'_0 \in \overline{X}_j] \leq l/(2t - 1) = l/(2t).
\]

We conclude by Lemma 2 that
\[
\mathbb{E}[\Pr_P[\mathcal{E}']\chi_j(P)] \leq \frac{2}{20s} \mathbb{E}[\Pr_P[\mathcal{E}']] \leq \frac{1}{2s} \mathbb{E}[\Pr_P[\mathcal{E}']] .
\]

By using the inequality from above we get
\[
\mathbb{E}[\Pr_P[\mathcal{E}']\chi(P)] \leq \mathbb{E}\left[\Pr_P[\mathcal{E}'] \sum_{j=1}^s \chi_j(P)\right] \leq \sum_{j=1}^s \mathbb{E}[\Pr_P[\mathcal{E}']\chi_j(P)] \leq \sum_{j=1}^s \frac{1}{2s} \mathbb{E}[\Pr_P[\mathcal{E}']] = \frac{1}{2} \mathbb{E}[\Pr_P[\mathcal{E}']]. \tag{4}
\]

Therefore by combining all inequalities we archive the desired result
\[
\Pr[\mathcal{E}] \overset{(3)}{\geq} \frac{1}{c} \mathbb{E}[\Pr_P[\mathcal{E}']] - \frac{1}{c} \mathbb{E}[\Pr_P[\mathcal{E}']\chi(P)] - s2^{-ct/s^3}
\]
\[
\overset{(4)}{>} \frac{1}{c} \mathbb{E}[\Pr_P[\mathcal{E}']] - \frac{1}{2c} \mathbb{E}[\Pr_P[\mathcal{E}']] - s2^{-ct/s^3}
\]
\[
= \frac{1}{2c} \mathbb{E}[\Pr_P[\mathcal{E}']] - s2^{-ct/s^3}.
\]

\[ \square \]

**Lemma 2.** There exists a choice for the constant \( c > 0 \) such that a partition \( P \) is \( j \)-bad and the following holds. For any set of \( s - 1 \) pairwise disjoint \( t \)-subsets \( I'_r \subset N \), \( 1 \leq r \leq s, r \neq j \), the conditional probability that the partition \( P = I_1 \cup \cdots \cup I_s \cup \{x\} \) is \( j \)-bad, given that \( I_r = I'_r \) for all \( r \neq j \), is at most \( \frac{1}{20s} \).
Proof. Due to the condition above the sets $I_i$ for $r \neq j$ and the union $I_j \cup \{x\}$ are known. Therefore we have $2t$ possibilities to construct a partition $P$ by choosing an element of the union as $x$. We consider a simple case distinction over the set

$$B = \{ C \subseteq I_j \cup \{x\} : C \in \mathcal{X}_j, |C| = t \}.$$  

The case $|B| < \frac{1}{2} (2^t)^{2^{-ct/s^3}}$ implies $\Pr_P [A_0^j \in \mathcal{X}_j] < 2^{-ct/s^3}$ for all possible partitions $P$. We proof this implication by contraposition.

$$\exists \text{ Partition } P : \Pr_P [A_0^j \in \mathcal{X}_j] \geq 2^{-ct/s^3} \implies \Pr_P [\text{chosen } t\text{-subset } \in \mathcal{X}_j] \geq 2^{-ct/s^3} \implies |B| / \left( \begin{array}{c} 2t \cr t \end{array} \right) \geq 2^{-ct/s^3} \implies |B| \geq 2^{-ct/s^3} \left( \begin{array}{c} 2t \cr t \end{array} \right) = \frac{1}{2} \left( \begin{array}{c} 2t \cr t \end{array} \right) 2^{-ct/s^3}.$$  

This in turn implies that no partition $P$ is $j$-bad by combining this implication and the definition of a $j$-bad partition $P$. We conclude that in this case the claim of this Lemma holds since the conditional probability is zero.

Now we consider the other case $|B| \geq \frac{1}{2} (2^t)^{2^{-ct/s^3}}$ and we denote by $\mathcal{F}$ the family of all subsets of $I_j \cup \{x\}$ of cardinality $t$. Furthermore we set $I_j \cup \{x\} = \{x_1, \ldots, x_{2t}\}$ and we denote by $p_i$ the probability that

$$p_i = \frac{|\{C \in \mathcal{F} : x_i \in C\}|}{|\mathcal{F}|}.$$  

Due to the binary entropy function and a standard inequality we can bound the size of $\mathcal{F}$

$$|\mathcal{F}| \leq 2^{\sum_{i=1}^{2t} H(p_i)}. \quad (5)$$  

Now we have to complete the partition $P$ by choosing a $x_i$ as $x$. We combine this with the question which $x_i$ results in a $j$-bad partition $P$ and we denote by $b$ the number of responsible $x_i$. Due to the definition of a $j$-bad partition $P$ we archive $p_i < (1 - 1/(s+1))(1 - p_i)$ which leads to an upper bound of $H(p_i)$. The first step is to bound the probability $p_i$ such that

$$p_i < (1 - \frac{1}{s+1})(1 - p_i) \iff p_i < \frac{2}{3} \leq \frac{s}{s+1} \iff p_i < \frac{2}{5}.$$  

By choosing a positive constant $c' \leq 2/25$ it follows

$$H(p_i) \leq 2\sqrt{p_i(1 - p_i)} < 0.98 \leq 1 - c'/s^2, s > 1.$$  

This fact is sufficient to give an upper bound to the size of $\mathcal{F}$ where we distinguish between $x_i$ which leads to a $j$-bad partition $P$ or not

$$|\mathcal{F}| \leq 2^{\sum_{i=1}^{2t} H(p_i)} = 2^{\sum_{j \text{ not } j\text{-bad}} H(p_i) + \sum_{j\text{ bad}} H(p_i)} \leq 2^{(2t - b) + b(1 - c'/s^2)} = 2^{2t - bc'/s^2}.$$  

Together with the lower bound of the size of $B$ from above we obtain

$$\frac{1}{2} \left( \begin{array}{c} 2t \cr t \end{array} \right) 2^{-ct/s^3} \leq |\mathcal{F}| \leq 2^{2t - bc'/s^2}. \quad (6)$$  

This again implies that
\[
\left( \frac{t}{s^3} \gg \log t \implies b \leq \frac{c_1 c t}{s} \right) \tag{7}
\]

where \(c_1 \geq 2/c'\) is a constant and \(t/s^3 \gg \log t\) means in this case there exists a sufficiently large enough constant \(a > 1\) such that \(t/s^3 > a \log t\) holds. We show that (6) implies (7) by contradiction. Therefore, we assume \(b > \frac{c_1 c t}{s}\) and it is sufficient to proof

\[
2t - bc'/s^2 + ct/s^3 + 1 \leq \left( \frac{2t}{t} \right).
\]

Starting from the left side of the inequality we obtain

\[
2t - bc'/s^2 + ct/s^3 + 1 \leq 2t - ct/s^3 (c_1 c' - 1) + 1 \leq 2t + 1 - ct/s^3 \leq 2t + 1 - ca \log t = \frac{2t + 1}{tca} \leq \frac{t2^t}{tca} = \frac{2^t}{tca - 1}.
\]

By choosing a sufficiently small constant \(c \geq 5/(2a)\) we get our desired contradiction

\[
\frac{2^t}{tca - 1} \leq \frac{2^t}{t^{3/2}} \leq \frac{2^t}{t^{1/2}} = \frac{2^{t-1}}{t^{1/2}} \leq \left( \frac{2t}{t} \right).
\]

The choice of \(c\) together with the other constants are enough to adapt the shown upper bound of \(b\) such that \(b \leq \frac{2^t}{2ca}\). Remember that \(b\) denotes the number of \(x_i\) which leads to a \(j\)-bad partition \(P\) and there exists only \(2t\) possibilities for a partition \(P\). By an average argument we obtain the required result. □

**Lemma 3.** If \(P = I_1 \cup \cdots \cup I_s \cup \{x\}\) is a good partition then

\[
\Pr_P\left[(A^1_1, \ldots, A^1_s) \in \bar{X}_1 \times \cdots \times \bar{X}_s\right] \geq \frac{1}{e} \Pr_P\left[(A^0_1, \ldots, A^0_s) \in \bar{X}_1 \times \cdots \times \bar{X}_s\right] - s 2^{-ct/s^3}.
\]

**Proof.** By negating the definition of a bad partition \(P\) we obtain a good partition such that

\[
\Pr_P[A^1_j \in \bar{X}_j] \geq \left(1 - \frac{1}{s + 1}\right) \Pr_P[A^0_j \in \bar{X}_j] - s 2^{-ct/s^3}, \forall j, 1 \leq j \leq s.
\]

The desired result is obtained by using the definition of the distribution \(\mu\), an upper bound of the inverse natural exponential function and the following fact

\[
(a - b)^n \geq a^n - na^{n-1}b. \tag{8}
\]

Therefore we get
\[
\Pr_P [(A_1^0 \times \cdots \times A_s^0) \in \overline{X}_1 \times \cdots \times \overline{X}_s] 
\geq \prod_{j=1}^{s} \left( (1 - \frac{1}{s+1}) \Pr_P [A_j^0 \in \overline{X}_j] - 2^{-ct/s^3} \right) 
\] 

\[ 
\geq \prod_{j=1}^{s} \left( (1 - \frac{1}{s+1}) \Pr_P [A_j^0 \in \overline{X}_j] \right) - \sum_{i=1}^{s} \left( \prod_{i \neq j}^{s} (1 - \frac{1}{s+1}) \Pr_P [A_j^0 \in \overline{X}_j] \right) 2^{-ct/s^3} 
\] 

\[ = \left( 1 - \frac{1}{s+1} \right)^s \Pr_P [(A_1^0 \times \cdots \times A_s^0) \in \overline{X}_1 \times \cdots \times \overline{X}_s] - s2^{-ct/s^3} 
\] 

\[ > \frac{1}{e} \Pr_P [(A_1^0 \times \cdots \times A_s^0) \in \overline{X}_1 \times \cdots \times \overline{X}_s] - s2^{-ct/s^3}. \]