Entropy and Shannon’s Theorem

Lecture 24
November 18, 2015
Part I

Entropy
Storing all strings of length $n$ and $j$ bits on

1. $S_{n,j}$: set of all strings of length $n$ with $j$ ones in them.
2. $T_{n,j}$: prefix tree storing all $S_{n,j}$.

![Diagram of prefix tree $T_{n,j}$ with nodes $T_{0,0}$, $T_{1,1}$, and $T_{1,0}$]
Binary strings of length 4
Binary strings of length 4

$S_{4,0} = \{0000\} \implies \#(0000) = 0.$
Binary strings of length 4

\[ S_{4,1} = \{0001, 0010, 0100, 1000\} \]
\[ \implies \#(0001) = 0. \]
\[ \#(0010) = 1. \]
\[ \#(0100) = 2. \]
\[ \#(1000) = 3. \]
Binary strings of length 4

\[ S_{4,2} = \{0011, 0101, 0110, 1001, 1010, 1100\} \]

\[ \#(0011) = 0. \]
\[ \#(0101) = 1. \]
\[ \#(0110) = 2. \]
\[ \#(1001) = 3. \]
\[ \#(1010) = 4. \]
\[ \#(1100) = 5. \]
Binary strings of length 4

$$S_{4,3} = \{0111, 1011, 1101, 1110\}$$

$$(0111) = 0.$$
$$(1011) = 1.$$
$$(1101) = 2.$$
$$(1110) = 3.$$
Binary strings of length 4

1. \( S_{4,4} = \{1111\} \)

\[ \Rightarrow \]

\( #(1111) = 0. \)

2.
Prefix tree \( \forall \) binary strings of length \( n \) with \( j \) ones

\[
T_{n,j} = \begin{cases} 
T_{n-1,j} & \text{if } j = 0 \\
T_{n-1,j-1} & \text{if } j > 0 
\end{cases}
\]

\[
T_{n,0} = \begin{cases} 
T_{n-1,0} & \text{if } j = 0 \\
0 & \text{if } j > 0 
\end{cases}
\]

\[
T_{n,n} = \begin{cases} 
T_{n-1,n-1} & \text{if } j = 0 \\
1 & \text{if } j > 0 
\end{cases}
\]
Prefix tree \forall binary strings of length \( n \) with \( j \) ones

\[
T_{n,j} = T_{n-1,j} + T_{n-1,j-1}
\]

# of leafs:

\[
|T_{n,j}| = |T_{n-1,j}| + |T_{n-1,j-1}|
\]
Prefix tree \( \forall \) binary strings of length \( n \) with \( j \) ones

\[
T_{n,j}
\]

\[|
T_{n,j} \rangle = |T_{n-1,j}| + |T_{n-1,j-1}|
\]

\[
(n) = (n-1) + (n-1)
\]

# of leafs:

\[
T_{n,0}
\]

\[
T_{n,n}
\]

---

Sariel (UIUC)  New CS473  6  Fall 2015  6 / 25
Prefix tree \( \forall \) binary strings of length \( n \) with \( j \) ones

\[ T_{n,j} \]

\begin{align*}
|T_{n,j}| &= |T_{n-1,j}| + |T_{n-1,j-1}| \\
\binom{n}{j} &= \binom{n-1}{j} + \binom{n-1}{j-1} \\
\implies |T_{n,j}| &= \binom{n}{j}.
\end{align*}
Encoding a string in $S_{n,j}$

1. $T_{n,j}$ leaves corresponds to strings of $S_{n,j}$.
2. Order all strings of $S_{n,j}$ order in lexicographical ordering.
3. $\equiv$ ordering leaves of $T_{n,j}$ from left to right.
4. Input: $s \in S_{n,j}$: compute index of $s$ in sorted set $S_{n,j}$.
5. $\text{EncodeBinomCoeff}(s)$ denote this polytime procedure.
Encoding a string in $S_{n,j}$

1. $T_{n,j}$ leafs corresponds to strings of $S_{n,j}$.
2. Order all strings of $S_{n,j}$ order in lexicographical ordering.
   $\equiv$ ordering leafs of $T_{n,j}$ from left to right.
3. Input: $s \in S_{n,j}$: compute index of $s$ in sorted set $S_{n,j}$.
4. $\text{EncodeBinomCoeff}(s)$ denote this polytime procedure.
Encoding a string in $S_{n,j}$

1. $T_{n,j}$ leafs corresponds to strings of $S_{n,j}$.
2. Order all strings of $S_{n,j}$ order in lexicographical ordering
3. $\equiv$ ordering leafs of $T_{n,j}$ from left to right.

Input: $s \in S_{n,j}$: compute index of $s$ in sorted set $S_{n,j}$.

4. $\text{EncodeBinomCoeff}(s)$ denote this polytime procedure.
Encoding a string in $S_{n,j}$

1. $T_{n,j}$ leaves corresponds to strings of $S_{n,j}$.
2. Order all strings of $S_{n,j}$ order in lexicographical ordering.
3. $\equiv$ ordering leaves of $T_{n,j}$ from left to right.

4. Input: $s \in S_{n,j}$: compute index of $s$ in sorted set $S_{n,j}$.
5. $\text{EncodeBinomCoeff}(s)$ denote this polytime procedure.
Encoding a string in $S_{n,j}$

1. $T_{n,j}$ leafs corresponds to strings of $S_{n,j}$.
2. Order all strings of $S_{n,j}$ order in lexicographical ordering.
3. $\equiv$ ordering leafs of $T_{n,j}$ from left to right.

4. Input: $s \in S_{n,j}$: compute index of $s$ in sorted set $S_{n,j}$.
5. $\text{EncodeBinomCoeff}(s)$ denote this polytime procedure.
Decoding a string in $S_{n,j}$

1. $T_{n,j}$ leafs corresponds to strings of $S_{n,j}$.
2. Order all strings of $S_{n,j}$ order in lexicographical ordering.
3. $\equiv$ ordering leafs of $T_{n,j}$ from left to right.
4. $x \in \left\{1, \ldots, \binom{n}{j}\right\}$: compute $x$th string in $S_{n,j}$ in polytime.
5. $\text{DecodeBinomCoeff}(x)$ denote this procedure.
Decoding a string in $S_{n,j}$

1. $T_{n,j}$ leafs correspond to strings of $S_{n,j}$.
2. Order all strings of $S_{n,j}$ order in lexicographical ordering.
3. $\equiv$ ordering leafs of $T_{n,j}$ from left to right.
4. $x \in \left\{1, \ldots, \binom{n}{j}\right\}$: compute $x$th string in $S_{n,j}$ in polytime.
5. $\text{DecodeBinomCoeff} (x)$ denote this procedure.
Decoding a string in $S_{n,j}$

1. $T_{n,j}$ leaves corresponds to strings of $S_{n,j}$.
2. Order all strings of $S_{n,j}$ order in lexicographical ordering.
3. $T_{n,j}$ ordering leafs of $T_{n,j}$ from left to right.

4. $x \in \{1, \ldots, \binom{n}{j}\}$: compute $x$th string in $S_{n,j}$ in polytime.
5. **DecodeBinomCoeff** ($x$) denote this procedure.
Decoding a string in $S_{n,j}$

1. $T_{n,j}$ leafs corresponds to strings of $S_{n,j}$.
2. Order all strings of $S_{n,j}$ order in lexicographical ordering.
3. $\equiv$ ordering leafs of $T_{n,j}$ from left to right.

\[
T_{n,j}
\]

\[
\begin{array}{c}
0 \\
T_{n-1,j} \\
T_{n-1,j} \\
1
\end{array}
\]

4. $x \in \{1, \ldots, \binom{n}{j}\}$: compute $x$th string in $S_{n,j}$ in polytime.

5. $\text{DecodeBinomCoeff}(x)$ denote this procedure.
Decoding a string in $S_{n,j}$

1. $T_{n,j}$ leafs corresponds to strings of $S_{n,j}$.
2. Order all strings of $S_{n,j}$ order in lexicographical ordering.
3. ≡ ordering leafs of $T_{n,j}$ from left to right.

4. $x \in \{1, \ldots, \binom{n}{j}\}$: compute $x$th string in $S_{n,j}$ in polytime.
5. DecodeBinomCoeff $(x)$ denote this procedure.
Encoding/decoding strings of $S_{n,j}$

**Lemma**

$S_{n,j}$: Set of binary strings of length $n$ with $j$ ones, sorted lexicographically.

1. **EncodeBinomCoeff**($\alpha$): Input is string $\alpha \in S_{n,j}$, compute index $x$ of $\alpha$ in $S_{n,j}$ in polynomial time in $n$.

2. **DecodeBinomCoeff**($x$): Input index $x \in \{1, \ldots, \binom{n}{j}\}$.

   Output $x$th string $\alpha$ in $S_{n,j}$, in time $O(\text{polylog } n + n)$. 
Theorem

Consider a coin that comes up heads with probability $p > 1/2$. For any constant $\delta > 0$ and for $n$ sufficiently large:

(A) One can extract, from an input of a sequence of $n$ flips, an output sequence of $(1 - \delta)n\mathbb{H}(p)$ (unbiased) independent random bits.

(B) One can not extract more than $n\mathbb{H}(p)$ bits from such a sequence.
Proof...

1. There are \( \binom{n}{j} \) input strings with exactly \( j \) heads.
2. Each has probability \( p^j (1 - p)^{n-j} \).
3. Map string \( s \) like that to index number in the set \( S_j = \{1, \ldots, \binom{n}{j}\} \).
4. Given that input string \( s \) has \( j \) ones (out of \( n \) bits) defines a uniform distribution on \( S_{n,j} \).
5. \( x \leftarrow \text{EncodeBinomCoeff}(s) \)
6. Uniform distributed in \( \{1, \ldots, N\} \), \( N = \binom{n}{j} \).
7. Seen in previous lecture...
8. ... extract in expectation, \( \lfloor \log N \rfloor - 1 \) bits from uniform random variable in the range \( 1, \ldots, N \).
9. Extract bits using \( \textbf{ExtractRandomness}(x, N) \).
Proof...

1. There are \( \binom{n}{j} \) input strings with exactly \( j \) heads.
2. Each has probability \( p^j(1 - p)^{n-j} \).
3. Map string \( s \) like that to index number in the set \( S_j = \{1, \ldots, \binom{n}{j}\} \).
4. Given that input string \( s \) has \( j \) ones (out of \( n \) bits) defines a uniform distribution on \( S_{n,j} \).
5. \( x \leftarrow \text{EncodeBinomCoeff}(s) \)
6. \( x \) uniform distributed in \( \{1, \ldots, N\} \), \( N = \binom{n}{j} \).
7. Seen in previous lecture...
8. ... extract in expectation, \( \lfloor \log N \rfloor - 1 \) bits from uniform random variable in the range 1, \ldots, \( N \).
9. Extract bits using \( \text{ExtractRandomness}(x, N) \).
Proof...

1. There are $\binom{n}{j}$ input strings with exactly $j$ heads.
2. Each has probability $p^j(1-p)^{n-j}$.
3. Map string $s$ like that to index number in the set $S_j = \{1, \ldots, \binom{n}{j}\}$.
4. Given that input string $s$ has $j$ ones (out of $n$ bits) defines a uniform distribution on $S_{n,j}$.
5. $x \leftarrow \text{EncodeBinomCoeff}(s)$
6. $x$ uniform distributed in $\{1, \ldots, N\}$, $N = \binom{n}{j}$.
7. Seen in previous lecture...
8. ... extract in expectation, $\lfloor \lg N \rfloor - 1$ bits from uniform random variable in the range $1, \ldots, N$.
9. Extract bits using $\text{ExtractRandomness}(x, N)$.
Proof...

1. There are \( \binom{n}{j} \) input strings with exactly \( j \) heads.
2. Each has probability \( p^j(1 - p)^{n-j} \).
3. Map string \( s \) like that to index number in the set \( S_j = \{1, \ldots, \binom{n}{j}\} \).
4. Given that input string \( s \) has \( j \) ones (out of \( n \) bits) defines a uniform distribution on \( S_{n,j} \).
5. \( x \leftarrow \text{EncodeBinomCoeff}(s) \)
6. \( x \) uniform distributed in \( \{1, \ldots, N\} \), \( N = \binom{n}{j} \).
7. Seen in previous lecture...
8. ... extract in expectation, \( \lceil \lg N \rceil - 1 \) bits from uniform random variable in the range \( 1, \ldots, N \).
9. Extract bits using \( \text{ExtractRandomness}(x, N) \).
Proof...

1. There are $\binom{n}{j}$ input strings with exactly $j$ heads.
2. Each has probability $p^j(1 - p)^{n-j}$.
3. Map string $s$ like that to index number in the set $S_j = \{1, \ldots, \binom{n}{j}\}$.
4. Given that input string $s$ has $j$ ones (out of $n$ bits) defines a uniform distribution on $S_{n,j}$.
5. $x \leftarrow \text{EncodeBinomCoeff}(s)$
6. $x$ uniform distributed in $\{1, \ldots, N\}$, $N = \binom{n}{j}$.
7. Seen in previous lecture...
8. ... extract in expectation, $\lfloor \lg N \rfloor - 1$ bits from uniform random variable in the range $1, \ldots, N$.
9. Extract bits using $\text{ExtractRandomness}(x, N)$.
Proof...

1. There are $\binom{n}{j}$ input strings with exactly $j$ heads.
2. Each has probability $p^j(1-p)^{n-j}$.
3. Map string $s$ like that to index number in the set $S_j = \{1, \ldots, \binom{n}{j}\}$.
4. Given that input string $s$ has $j$ ones (out of $n$ bits) defines a uniform distribution on $S_{n,j}$.
5. $x \leftarrow \text{EncodeBinomCoeff}(s)$
6. $x$ uniform distributed in $\{1, \ldots, N\}$, $N = \binom{n}{j}$.
7. Seen in previous lecture...
8. ... extract in expectation, $\lceil \lg N \rceil - 1$ bits from uniform random variable in the range $1, \ldots, N$.
9. Extract bits using $\text{ExtractRandomness}(x, N)$.
Proof...

1. There are \( \binom{n}{j} \) input strings with exactly \( j \) heads.
2. Each has probability \( p^j(1 - p)^{n-j} \).
3. Map string \( s \) like that to index number in the set \( S_j = \{1, \ldots, \binom{n}{j}\} \).
4. Given that input string \( s \) has \( j \) ones (out of \( n \) bits) defines a uniform distribution on \( S_{n,j} \).
5. \( x \leftarrow \text{EncodeBinomCoeff}(s) \)
6. \( x \) uniform distributed in \( \{1, \ldots, N\} \), \( N = \binom{n}{j} \).
7. Seen in previous lecture...
8. ... extract in expectation, \( \lfloor \lg N \rfloor - 1 \) bits from uniform random variable in the range \( 1, \ldots, N \).
9. Extract bits using \( \text{ExtractRandomness}(x, N) \).
Proof...

1. There are $\binom{n}{j}$ input strings with exactly $j$ heads.
2. Each has probability $p^j(1 - p)^{n-j}$.
3. Map string $s$ like that to index number in the set $S_j = \{1, \ldots, \binom{n}{j}\}$.
4. Given that input string $s$ has $j$ ones (out of $n$ bits) defines a uniform distribution on $S_{n,j}$.
5. $x \leftarrow \text{EncodeBinomCoeff}(s)$
6. $x$ uniform distributed in $\{1, \ldots, N\}$, $N = \binom{n}{j}$.
7. Seen in previous lecture...
8. ... extract in expectation, $\lfloor \log N \rfloor - 1$ bits from uniform random variable in the range $1, \ldots, N$.
9. Extract bits using $\text{ExtractRandomness}(x, N)$.
Proof...

1. There are \( \binom{n}{j} \) input strings with exactly \( j \) heads.

2. Each has probability \( p^j(1 - p)^{n-j} \).

3. Map string \( s \) like that to index number in the set \( S_j = \{1, \ldots, \binom{n}{j}\} \).

4. Given that input string \( s \) has \( j \) ones (out of \( n \) bits) defines a uniform distribution on \( S_{n,j} \).

5. \( x \leftarrow \text{EncodeBinomCoeff}(s) \)

6. \( x \) uniform distributed in \( \{1, \ldots, N\} \), \( N = \binom{n}{j} \).

7. Seen in previous lecture...

8. ... extract in expectation, \( \lfloor \lg N \rfloor - 1 \) bits from uniform random variable in the range \( 1, \ldots, N \).

9. Extract bits using \( \text{ExtractRandomness}(x, N) \).
Exciting proof continued...

1. $Z$: random variable: number of heads in input string $s$.
2. $B$: number of random bits extracted.
3. \[
   E[B] = \sum_{k=0}^{n} \Pr[Z = k] \cdot E[B \mid Z = k],
\]
4. Know: $E[B \mid Z = k] \geq \left\lfloor \log \left( \frac{n}{k} \right) \right\rfloor - 1$.
5. $\varepsilon < p - 1/2$: sufficiently small constant.
6. $n(p - \varepsilon) \leq k \leq n(p + \varepsilon)$:
   \[
   \binom{n}{k} \geq \binom{n}{\left\lfloor n(p + \varepsilon) \right\rfloor} \geq \frac{2^{nH(p+\varepsilon)}}{n + 1},
   \]
7. ... since $2^{nH(p)}$ is a good approximation to $\binom{n}{np}$ as proved in previous lecture.
Exciting proof continued...

1. \( Z \): random variable: number of heads in input string \( s \).
2. \( B \): number of random bits extracted.

\[
E[B] = \sum_{k=0}^{n} \Pr[Z = k] E[B \mid Z = k],
\]

3. Know: \( E[B \mid Z = k] \geq \left\lfloor \lg \left( \binom{n}{k} \right) \right\rfloor - 1. \)

4. \( \varepsilon < p - 1/2 \): sufficiently small constant.

5. \( n(p - \varepsilon) \leq k \leq n(p + \varepsilon) \):

\[
\binom{n}{k} \geq \left\lfloor n(p + \varepsilon) \right\rfloor \geq \frac{2^{n\mathbb{H}(p+\varepsilon)}}{n + 1},
\]

6. ... since \( 2^{n\mathbb{H}(p)} \) is a good approximation to \( \binom{n}{np} \) as proved in previous lecture.
Z: random variable: number of heads in input string $s$.

$B$: number of random bits extracted.

$$\mathbb{E}[B] = \sum_{k=0}^{n} \Pr[Z = k] \mathbb{E}[B \mid Z = k],$$

Know: $\mathbb{E}[B \mid Z = k] \geq \left\lceil \log \left( \frac{n}{k} \right) \right\rceil - 1$.

$\varepsilon < p - 1/2$: sufficiently small constant.

$n(p - \varepsilon) \leq k \leq n(p + \varepsilon)$:

$$\binom{n}{k} \geq \binom{n}{\lfloor n(p + \varepsilon) \rfloor} \geq \frac{2^{n \mathbb{H}(p+\varepsilon)}}{n+1},$$

... since $2^{n \mathbb{H}(p)}$ is a good approximation to $\binom{n}{np}$ as proved in previous lecture.
Exciting proof continued...

1. \( Z \): random variable: number of heads in input string \( s \).
2. \( B \): number of random bits extracted.

\[
E[B] = \sum_{k=0}^{n} \Pr[Z = k] E[B | Z = k],
\]

3. Know: \( E[B | Z = k] \geq \left\lfloor \log \binom{n}{k} \right\rfloor - 1 \).
4. \( \epsilon < p - 1/2 \): sufficiently small constant.
5. \( n(p - \epsilon) \leq k \leq n(p + \epsilon) \):

\[
\binom{n}{k} \geq \binom{n}{\lfloor n(p + \epsilon) \rfloor} \geq \frac{2^{nH(p+\epsilon)}}{n+1},
\]

6. ... since \( 2^{nH(p)} \) is a good approximation to \( \binom{n}{np} \) as proved in previous lecture.
Z: random variable: number of heads in input string $s$.

$B$: number of random bits extracted.

$$E[B] = \sum_{k=0}^{n} \Pr[Z = k] E[B \mid Z = k],$$

Know: $E[B \mid Z = k] \geq \left\lfloor \log \left( \binom{n}{k} \right) \right\rfloor - 1$.

$\varepsilon < p - 1/2$: sufficiently small constant.

$n(p - \varepsilon) \leq k \leq n(p + \varepsilon)$:

$$\binom{n}{k} \geq \left( \left\lfloor n(p + \varepsilon) \right\rfloor \right) \geq \frac{2^{n\mathbb{H}(p+\varepsilon)}}{n+1},$$

... since $2^{n\mathbb{H}(p)}$ is a good approximation to $\binom{n}{np}$ as proved in previous lecture.
1. \( Z \): random variable: number of heads in input string \( s \).

2. \( B \): number of random bits extracted.

\[
E[B] = \sum_{k=0}^{n} \Pr[Z = k] E[B \mid Z = k],
\]

3. Know: \( E[B \mid Z = k] \geq \lceil \log \left( \binom{n}{k} \right) \rceil - 1. \)

4. \( \varepsilon < p - 1/2 \): sufficiently small constant.

5. \( n(p - \varepsilon) \leq k \leq n(p + \varepsilon): \)

\[
\binom{n}{k} \geq \binom{n}{\lfloor n(p + \varepsilon) \rfloor} \geq \frac{2^n \mathbb{H}(p + \varepsilon)}{n + 1},
\]

6. ... since \( 2^n \mathbb{H}(p) \) is a good approximation to \( \binom{n}{np} \) as proved in previous lecture.
Exciting proof continued...

1. \( Z \): random variable: number of heads in input string \( s \).
2. \( B \): number of random bits extracted.

\[
E[B] = \sum_{k=0}^{n} \Pr[Z = k] E[B \mid Z = k],
\]

3. Know: \( E[B \mid Z = k] \geq \left\lfloor \log \left( \frac{n}{k} \right) \right\rfloor - 1. \)
4. \( \varepsilon < p - 1/2 \): sufficiently small constant.
5. \( n(p - \varepsilon) \leq k \leq n(p + \varepsilon) \):

\[
\binom{n}{k} \geq \left\lfloor n(p + \varepsilon) \right\rfloor \geq \frac{2^{n\mathbb{H}(p+\varepsilon)}}{n + 1},
\]

6. \( 2^{n\mathbb{H}(p)} \) is a good approximation to \( \binom{n}{np} \) as proved in previous lecture.
Super exciting proof continued...

\[ E[B] = \sum_{k=0}^{n} \Pr[Z = k] E[B \mid Z = k]. \]

\[ E[B] \geq \sum_{k=\lceil n(p-\varepsilon) \rceil}^{\lfloor n(p+\varepsilon) \rfloor} \Pr[Z = k] \left( \left\lfloor \log \binom{n}{k} \right\rfloor - 1 \right) \]

\[ \geq \sum_{k=\lfloor n(p-\varepsilon) \rfloor}^{\lfloor n(p+\varepsilon) \rfloor} \Pr[Z = k] \left( \log \frac{2^{nH(p+\varepsilon)}}{n+1} - 2 \right) \]

\[ = \left( nH(p+\varepsilon) - \log(n+1) - 2 \right) \Pr[|Z - np| \leq \varepsilon n] \]

\[ \geq \left( nH(p+\varepsilon) - \log(n+1) - 2 \right) \left( 1 - 2 \exp \left( -\frac{n\varepsilon^2}{4p} \right) \right), \]

since \( \mu = E[Z] = np \) and

\[ \Pr[|Z - np| \geq \frac{\varepsilon}{p} pn] \leq 2 \exp \left( -\frac{np}{4} \left( \frac{\varepsilon}{p} \right)^2 \right) = 2 \exp \left( -\frac{n\varepsilon^2}{4p} \right), \]
Super exciting proof continued...

\[ E[B] = \sum_{k=0}^{n} \Pr[Z = k] E[B | Z = k]. \]

\[ E[B] \geq \sum_{k=\lceil n(p-\varepsilon) \rceil}^{\lceil n(p+\varepsilon) \rceil} \Pr[Z = k] \left( \lceil \log \binom{n}{k} \rceil - 1 \right) \]

\[ \geq \sum_{k=\lfloor n(p-\varepsilon) \rfloor}^{\lfloor n(p+\varepsilon) \rfloor} \Pr[Z = k] \left( \log \frac{2^{n\mathbb{H}(p+\varepsilon)}}{n+1} - 2 \right) \]

\[ = \left( n\mathbb{H}(p + \varepsilon) - \log(n + 1) - 2 \right) \Pr[|Z - np| \leq \varepsilon n] \]

\[ \geq \left( n\mathbb{H}(p + \varepsilon) - \log(n + 1) - 2 \right) \left( 1 - 2 \exp \left( -\frac{n\varepsilon^2}{4p} \right) \right), \]

since \( \mu = E[Z] = np \) and

\[ \Pr[|Z - np| \geq \frac{\varepsilon}{p} pn] \leq 2 \exp \left( -\frac{np}{4} \left( \frac{\varepsilon}{p} \right)^2 \right) = 2 \exp \left( -\frac{n\varepsilon^2}{4p} \right), \]
Super exciting proof continued...

\[ \mathbb{E}[B] = \sum_{k=0}^{n} \Pr[Z = k] \mathbb{E}[B \mid Z = k]. \]

\[ \mathbb{E}[B] \geq \sum_{k=\lceil n(p-\varepsilon) \rceil}^{\lceil n(p+\varepsilon) \rceil} \Pr[Z = k] \left( \left\lfloor \log \left( \frac{n}{k} \right) \right\rfloor - 1 \right) \]

\[ \geq \sum_{k=\lceil n(p-\varepsilon) \rceil}^{\lceil n(p+\varepsilon) \rceil} \Pr[Z = k] \left( \log \frac{2^{n\mathbb{H}(p+\varepsilon)}}{n+1} - 2 \right) \]

\[ = \left( n\mathbb{H}(p + \varepsilon) - \log(n + 1) - 2 \right) \Pr[|Z - np| \leq \varepsilon n] \]

\[ \geq \left( n\mathbb{H}(p + \varepsilon) - \log(n + 1) - 2 \right) \left( 1 - 2 \exp \left( -\frac{n\varepsilon^2}{4p} \right) \right), \]

since \( \mu = \mathbb{E}[Z] = np \) and

\[ \Pr[|Z - np| \geq \frac{\varepsilon}{p} pn] \leq 2 \exp \left( -\frac{np}{4} \left( \frac{\varepsilon}{p} \right)^2 \right) = 2 \exp \left( -\frac{n\varepsilon^2}{4p} \right), \]
Super exciting proof continued...

\[ E[B] = \sum_{k=0}^{n} \Pr[Z = k] E[B \mid Z = k]. \]

\[ E[B] \geq \sum_{k=\lceil n(p-\epsilon) \rceil}^{\lfloor n(p+\epsilon) \rfloor} \Pr[Z = k] E[B \mid Z = k]. \]

\[ \geq \sum_{k=\lfloor n(p-\epsilon) \rfloor}^{\lceil n(p+\epsilon) \rceil} \Pr[Z = k] \left( \left\lfloor \log \left( \frac{n}{k} \right) \right\rfloor - 1 \right) \]

\[ = \left( n \mathbb{H}(p + \epsilon) - \log(n + 1) - 2 \right) \Pr[|Z - np| \leq \epsilon n] \]

\[ \geq \left( n \mathbb{H}(p + \epsilon) - \log(n + 1) - 2 \right) \left( 1 - 2 \exp \left( -\frac{n\epsilon^2}{4p} \right) \right), \]

since \( \mu = E[Z] = np \) and

\[ \Pr[|Z - np| \geq \frac{\epsilon}{p} pn] \leq 2 \exp \left( -\frac{np}{4} \left( \frac{\epsilon}{p} \right)^2 \right) = 2 \exp \left( -\frac{n\epsilon^2}{4p} \right), \]
Super exciting proof continued...

\[ E[B] = \sum_{k=0}^{n} \Pr[Z = k] E[B \mid Z = k]. \]

\[ E[B] \geq \sum_{k=\lceil n(p-\varepsilon) \rceil}^{\lceil n(p+\varepsilon) \rceil} \Pr[Z = k] \left( \left\lfloor \frac{n}{k} \right\rfloor - 1 \right) \]

\[ \geq \sum_{k=\lceil n(p-\varepsilon) \rceil}^{\lceil n(p+\varepsilon) \rceil} \Pr[Z = k] \left( \lg \frac{2^{nH(p+\varepsilon)}}{n+1} - 2 \right) \]

\[ \geq \left( nH(p + \varepsilon) - \lg(n + 1) - 2 \right) \Pr[|Z - np| \leq \varepsilon n] \]

\[ \geq \left( nH(p + \varepsilon) - \lg(n + 1) - 2 \right) \left( 1 - 2 \exp \left( -\frac{n\varepsilon^2}{4p} \right) \right), \]

since \( \mu = E[Z] = np \) and

\[ \Pr[|Z - np| \geq \frac{\varepsilon}{p} pn] \leq 2 \exp \left( -\frac{np/4}{\varepsilon^2} \right) = 2 \exp \left( -\frac{n\varepsilon^2}{4p} \right), \]
Super exciting proof continued...

\[ E[B] = \sum_{k=0}^{n} \Pr[Z = k] E[B \mid Z = k]. \]

\[ E[B] \geq \sum_{k=[n(p-\varepsilon)]}^{\lceil n(p+\varepsilon) \rceil} \Pr[Z = k] E[B \mid Z = k] \]

\[ \geq \sum_{k=[n(p-\varepsilon)]}^{\lceil n(p+\varepsilon) \rceil} \Pr[Z = k] \left( \lg \left( \frac{n}{k} \right) - 1 \right) \]

\[ \geq \sum_{k=[n(p-\varepsilon)]}^{\lceil n(p+\varepsilon) \rceil} \Pr[Z = k] \left( \lg \frac{2^n H(p+\varepsilon)}{n+1} - 2 \right) \]

\[ = \left( n H(p + \varepsilon) - \lg(n + 1) - 2 \right) \Pr[|Z - np| \leq \varepsilon n] \]

\[ \geq \left( n H(p + \varepsilon) - \lg(n + 1) - 2 \right) \left( 1 - 2 \exp \left( -\frac{n\varepsilon^2}{4p} \right) \right), \]

since \( \mu = E[Z] = np \) and

\[ \Pr[|Z - np| \geq \frac{\varepsilon}{p} pn] \leq 2 \exp \left( -\frac{np}{4} \left( \frac{\varepsilon}{p} \right)^2 \right) = 2 \exp \left( -\frac{n\varepsilon^2}{4p} \right), \]
Super exciting proof continued...

\[
\mathbb{E}[B] = \sum_{k=0}^{n} \Pr[Z = k] \mathbb{E}[B \mid Z = k].
\]

\[
\mathbb{E}[B] \geq \sum_{k=\lceil n(p-\varepsilon) \rceil}^{\lceil n(p+\varepsilon) \rceil} \Pr[Z = k] \mathbb{E}[B \mid Z = k]
\]

\[
\geq \sum_{k=\lceil n(p-\varepsilon) \rceil}^{\lceil n(p+\varepsilon) \rceil} \Pr[Z = k] \left( \log \left( \frac{n}{k} \right) - 1 \right)
\]

\[
= \left( n \mathcal{H}(p + \varepsilon) - \log(n + 1) - 2 \right) \Pr[|Z - np| \leq \varepsilon n]
\]

\[
\geq \left( n \mathcal{H}(p + \varepsilon) - \log(n + 1) - 2 \right) \left( 1 - 2 \exp \left( -\frac{n\varepsilon^2}{4p} \right) \right),
\]

since \( \mu = \mathbb{E}[Z] = np \) and

\[
\Pr[|Z - np| \geq \frac{\varepsilon}{p} pn] \leq 2 \exp \left( -\frac{np}{4} \left( \frac{\varepsilon}{p} \right)^2 \right) = 2 \exp \left( -\frac{n\varepsilon^2}{4p} \right),
\]
1. Fix $\varepsilon > 0$, such that $\mathbb{H}(p + \varepsilon) > (1 - \delta/4)\mathbb{H}(p)$, $p$ is fixed.

2. $\implies n\mathbb{H}(p) = \Omega(n)$.

3. For $n$ sufficiently large: $-\log(n + 1) \geq -\frac{\delta}{10} n\mathbb{H}(p)$.

4. ... also $2 \exp\left(-\frac{n\varepsilon^2}{4p}\right) \leq \frac{\delta}{10}$.

5. For $n$ large enough;

$$E[B] \geq \left(1 - \frac{\delta}{4} - \frac{\delta}{10}\right) n\mathbb{H}(p) \left(1 - \frac{\delta}{10}\right) \geq (1 - \delta) n\mathbb{H}(p),$$
Fix $\varepsilon > 0$, such that $\mathbb{H}(p + \varepsilon) > (1 - \delta/4)\mathbb{H}(p)$, $p$ is fixed.

$\implies n\mathbb{H}(p) = \Omega(n),$

For $n$ sufficiently large: $-\lg(n + 1) \geq -\frac{\delta}{10} n\mathbb{H}(p)$.

... also $2 \exp\left(-\frac{n\varepsilon^2}{4p}\right) \leq \frac{\delta}{10}$.

For $n$ large enough:

$$
E[B] \geq \left(1 - \frac{\delta}{4} - \frac{\delta}{10}\right) n\mathbb{H}(p) \left(1 - \frac{\delta}{10}\right) \\
\geq (1 - \delta) n\mathbb{H}(p),
$$
Fix $\varepsilon > 0$, such that $\mathbb{H}(p + \varepsilon) > (1 - \delta/4)\mathbb{H}(p)$, $p$ is fixed.

$\mathbb{H}(p)$

$n\mathbb{H}(p) = \Omega(n),$

For $n$ sufficiently large: $-\log(n + 1) \geq -\frac{\delta}{10}n\mathbb{H}(p)$.

... also $2 \exp\left(-\frac{n\varepsilon^2}{4p}\right) \leq \frac{\delta}{10}$.

For $n$ large enough:

$$E[B] \geq \left(1 - \frac{\delta}{4} - \frac{\delta}{10}\right)n\mathbb{H}(p)\left(1 - \frac{\delta}{10}\right)$$

$$\geq (1 - \delta)n\mathbb{H}(p),$$
Fix $\varepsilon > 0$, such that $\mathbb{H}(p + \varepsilon) > (1 - \delta/4)\mathbb{H}(p)$, $p$ is fixed.

$\implies n\mathbb{H}(p) = \Omega(n)$.

For $n$ sufficiently large: $-\log(n + 1) \geq -\frac{\delta}{10} n\mathbb{H}(p)$.

... also $2 \exp\left(-\frac{n\varepsilon^2}{4p}\right) \leq \frac{\delta}{10}$.

For $n$ large enough;

$$
\mathbb{E}[B] \geq \left(1 - \frac{\delta}{4} - \frac{\delta}{10}\right) n\mathbb{H}(p) \left(1 - \frac{\delta}{10}\right) \\
\geq (1 - \delta) n\mathbb{H}(p),
$$
Fix $\varepsilon > 0$, such that $\mathbb{H}(p + \varepsilon) > (1 - \delta/4)\mathbb{H}(p)$, $p$ is fixed.

$$n\mathbb{H}(p) = \Omega(n),$$

For $n$ sufficiently large: $-\lg(n + 1) \geq -\frac{\delta}{10} n\mathbb{H}(p)$.

... also $2 \exp\left(-\frac{n\varepsilon^2}{4p}\right) \leq \frac{\delta}{10}$.

For $n$ large enough;

$$E[B] \geq \left(1 - \frac{\delta}{4} - \frac{\delta}{10}\right)n\mathbb{H}(p) \left(1 - \frac{\delta}{10}\right) \geq (1 - \delta)n\mathbb{H}(p),$$
Hyper super duper exciting proof continued...

1. Need to prove upper bound.

2. If input sequence $x$ has probability $\Pr[X = x]$, then $y = \text{Ext}(x)$ has probability to be generated $\geq \Pr[X = x]$.

3. All sequences of length $|y|$ have equal probability to be generated (by definition).

4. $2^{|\text{Ext}(x)|} \Pr[X = x] \leq 2^{|\text{Ext}(x)|} \Pr[y = \text{Ext}(x)] \leq 1$.

5. $\implies |\text{Ext}(x)| \leq \lg(1/\Pr[X = x])$

6. $E[B] = \sum_x \Pr[X = x] |\text{Ext}(x)|$

   $\leq \sum_x \Pr[X = x] \lg \frac{1}{\Pr[X = x]} = \mathbb{H}(X)$. 

\[\]
1. Need to prove upper bound.
2. If input sequence \( x \) has probability \( \Pr[X = x] \), then \( y = \text{Ext}(x) \) has probability to be generated \( \geq \Pr[X = x] \).
3. All sequences of length \(|y|\) have equal probability to be generated (by definition).
4. \( 2^{|\text{Ext}(x)|} \Pr[X = x] \leq 2^{|\text{Ext}(x)|} \Pr[y = \text{Ext}(x)] \leq 1. \)
5. \( \implies |\text{Ext}(x)| \leq \lg(1/\Pr[X = x]) \)
6. \( \mathbb{E}[B] = \sum_x \Pr[X = x] |\text{Ext}(x)| \leq \sum_x \Pr[X = x] \lg \frac{1}{\Pr[X = x]} = H(X). \)
1. Need to prove upper bound.
2. If input sequence $x$ has probability $\Pr[X = x]$, then $y = \text{Ext}(x)$ has probability to be generated $\geq \Pr[X = x]$.
3. All sequences of length $|y|$ have equal probability to be generated (by definition).
4. $2^{|\text{Ext}(x)|} \Pr[X = x] \leq 2^{|\text{Ext}(x)|} \Pr[y = \text{Ext}(x)] \leq 1$.
5. $\iff |\text{Ext}(x)| \leq \log(1/\Pr[X = x])$
6. $E[B] = \sum_x \Pr[X = x] |\text{Ext}(x)|$
   $\leq \sum_x \Pr[X = x] \log \frac{1}{\Pr[X = x]} = H(X)$. 

Need to prove upper bound. 

If input sequence $x$ has probability $\Pr[X = x]$, then $y = \text{Ext}(x)$ has probability to be generated $\geq \Pr[X = x]$. 

All sequences of length $|y|$ have equal probability to be generated (by definition). 

$2^{|\text{Ext}(x)|} \Pr[X = x] \leq 2^{|\text{Ext}(x)|} \Pr[y = \text{Ext}(x)] \leq 1$. 

$\implies |\text{Ext}(x)| \leq \lg(1/\Pr[X = x])$ 

$E[B] = \sum_x \Pr[X = x] |\text{Ext}(x)| 
\leq \sum_x \Pr[X = x] \lg \frac{1}{\Pr[X = x]} = \mathbb{H}(X)$. 

Need to prove upper bound.

If input sequence $x$ has probability $\Pr[X = x]$, then $y = \text{Ext}(x)$ has probability to be generated $\geq \Pr[X = x]$.

All sequences of length $|y|$ have equal probability to be generated (by definition).

$$2^{|\text{Ext}(x)|} \Pr[X = x] \leq 2^{|\text{Ext}(x)|} \Pr[y = \text{Ext}(x)] \leq 1.$$ 

$$\implies |\text{Ext}(x)| \leq \lg(1/\Pr[X = x])$$

$$\mathbb{E}[B] = \sum_x \Pr[X = x] |\text{Ext}(x)|$$

$$\leq \sum_x \Pr[X = x] \lg \frac{1}{\Pr[X = x]} = \mathbb{H}(X).$$
Need to prove upper bound.

If input sequence $x$ has probability $\Pr[X = x]$, then $y = \text{Ext}(x)$ has probability to be generated $\geq \Pr[X = x]$.

All sequences of length $|y|$ have equal probability to be generated (by definition).

$2^{|\text{Ext}(x)|} \Pr[X = x] \leq 2^{|\text{Ext}(x)|} \Pr[y = \text{Ext}(x)] \leq 1$.

$\implies |\text{Ext}(x)| \leq \lg(1/\Pr[X = x])$

$\mathbb{E}[B] = \sum_x \Pr[X = x] |\text{Ext}(x)|$

$\leq \sum_x \Pr[X = x] \lg \frac{1}{\Pr[X = x]} = \mathbb{H}(X)$. 
1. Need to prove upper bound.
2. If input sequence $x$ has probability $Pr[X = x]$, then $y = Ext(x)$ has probability to be generated $\geq Pr[X = x]$.
3. All sequences of length $|y|$ have equal probability to be generated (by definition).
4. $2^{|Ext(x)|} Pr[X = x] \leq 2^{|Ext(x)|} Pr[y = Ext(x)] \leq 1$.
5. $\implies |Ext(x)| \leq \lg(1/Pr[X = x])$
6. $E[B] = \sum_x Pr[X = x] |Ext(x)|$

$$\leq \sum_x Pr[X = x] \lg \frac{1}{Pr[X = x]} = H(X).$$
1. Need to prove upper bound.

2. If input sequence $x$ has probability $\Pr[X = x]$, then $y = \text{Ext}(x)$ has probability to be generated $\geq \Pr[X = x]$.

3. All sequences of length $|y|$ have equal probability to be generated (by definition).

4. $2^{|\text{Ext}(x)|} \Pr[X = x] \leq 2^{|\text{Ext}(x)|} \Pr[y = \text{Ext}(x)] \leq 1$.

5. $\implies |\text{Ext}(x)| \leq \lg(1/\Pr[X = x])$

6. $E[B] = \sum_x \Pr[X = x] |\text{Ext}(x)|$
   $\leq \sum_x \Pr[X = x] \lg \frac{1}{\Pr[X = x]} = \mathbb{H}(X)$. 
Need to prove upper bound.

If input sequence $x$ has probability $\Pr[X = x]$, then $y = \text{Ext}(x)$ has probability to be generated $\geq \Pr[X = x]$. All sequences of length $|y|$ have equal probability to be generated (by definition).

$2^{|\text{Ext}(x)|} \Pr[X = x] \leq 2^{|\text{Ext}(x)|} \Pr[y = \text{Ext}(x)] \leq 1$.

$\implies |\text{Ext}(x)| \leq \lg(1/\Pr[X = x])$

$E[B] = \sum_x \Pr[X = x] |\text{Ext}(x)|$

$\leq \sum_x \Pr[X = x] \lg \frac{1}{\Pr[X = x]} = \mathbb{H}(X)$. 

$\blacksquare$
Part III

Coding: Shannon's Theorem
Shannon’s Theorem

**Definition**

1. **binary symmetric channel** with parameter $p$
2. sequence of bits $x_1, x_2, \ldots$, an
3. output: $y_1, y_2, \ldots$,
   a sequence of bits such that...
4. $\Pr[x_i = y_i] = 1 - p$ independently for each $i$. 
Shannon’s Theorem

Definition

1. **binary symmetric channel** with parameter \( p \)
2. sequence of bits \( x_1, x_2, \ldots \), an
3. output: \( y_1, y_2, \ldots \),
   a sequence of bits such that...
4. \( \Pr[x_i = y_i] = 1 - p \) independently for each \( i \).
Shannon’s Theorem

Definition

1. **binary symmetric channel** with parameter $p$
2. sequence of bits $x_1, x_2, \ldots$, an
3. output: $y_1, y_2, \ldots$,
   a sequence of bits such that...
4. $\Pr[x_i = y_i] = 1 - p$ independently for each $i$. 
**Shannon’s Theorem**

**Definition**

1. **binary symmetric channel** with parameter $p$
2. sequence of bits $x_1, x_2, \ldots$, an
3. output: $y_1, y_2, \ldots$,
   a sequence of bits such that...
4. $\Pr[x_i = y_i] = 1 - p$ independently for each $i$. 

$\Pr[x_i = y_i] = 1 - p$ independently for each $i$. 

Sariel (UIUC)  
New CS473 17  
Fall 2015  17 / 25
Encoding/decoding with noise

Definition

1. **Encoding function** $\text{Enc} : \{0, 1\}^k \rightarrow \{0, 1\}^n$ takes as input a sequence of $k$ bits and outputs a sequence of $n$ bits.

2. **Decoding function** $\text{Dec} : \{0, 1\}^n \rightarrow \{0, 1\}^k$ takes as input a sequence of $n$ bits and outputs a sequence of $k$ bits.
## Encoding/decoding with noise

### Definition

1. **$(k, n)$ encoding function** $\text{Enc} : \{0, 1\}^k \rightarrow \{0, 1\}^n$ takes as input a sequence of $k$ bits and outputs a sequence of $n$ bits.

2. **$(k, n)$ decoding function** $\text{Dec} : \{0, 1\}^n \rightarrow \{0, 1\}^k$ takes as input a sequence of $n$ bits and outputs a sequence of $k$ bits.
Claude Elwood Shannon (April 30, 1916 - February 24, 2001), an American electrical engineer and mathematician, has been called “the father of information theory”. His master thesis was how to building boolean circuits for any boolean function.
For a binary symmetric channel with parameter $p < 1/2$ and for any constants $\delta, \gamma > 0$, where $n$ is sufficiently large, the following holds:

(i) For an $k \leq n(1 - H(p) - \delta)$ there exists $(k, n)$ encoding and decoding functions such that the probability the receiver fails to obtain the correct message is at most $\gamma$ for every possible $k$-bit input messages.

(ii) There are no $(k, n)$ encoding and decoding functions with $k \geq n(1 - H(p) + \delta)$ such that the probability of decoding correctly is at least $\gamma$ for a $k$-bit input message chosen uniformly at random.
When the sender sends a string...

\[ S = s_1 s_2 \ldots s_n \]
When the sender sends a string...

\[ S = s_1 s_2 \ldots s_n \]
When the sender sends a string...

\[ S = s_1 s_2 \ldots s_n \]
When the sender sends a string...

\[ S = s_1 s_2 \ldots s_n \]
When the sender sends a string...

\[ S = s_1s_2 \ldots s_n \]

One ring to rule them all!
Some intuition...

1. senders sent string $S = s_1 s_2 \ldots s_n$.
2. receiver got string $T = t_1 t_2 \ldots t_n$.
3. $p = \Pr[t_i \neq s_i]$, for all $i$.
4. $U$: Hamming distance between $S$ and $T$: $U = \sum_i[s_i \neq t_i]$.
5. By assumption: $E[U] = pn$, and $U$ is a binomial variable.
6. By Chernoff inequality: $U \in [(1 - \delta)np, (1 + \delta)np]$ with high probability, where $\delta$ is a tiny constant.
7. $T$ is in a ring $R$ centered at $S$, with inner radius $(1 - \delta)np$ and outer radius $(1 + \delta)np$.
8. This ring has

$$\sum_{i=(1-\delta)np}^{(1+\delta)np} \binom{n}{i} \leq 2^{\binom{n}{(1+\delta)np}} \leq \alpha = 2 \cdot 2^n \mathbb{H}((1+\delta)p).$$

strings in it.
Some intuition...

1. senders sent string $S = s_1 s_2 \ldots s_n$.
2. receiver got string $T = t_1 t_2 \ldots t_n$.
3. $p = \Pr[t_i \neq s_i]$, for all $i$.
4. $U$: Hamming distance between $S$ and $T$: $U = \sum_i[s_i \neq t_i]$.
5. By assumption: $\mathbb{E}[U] = pn$, and $U$ is a binomial variable.
6. By Chernoff inequality: $U \in [(1 - \delta)np, (1 + \delta)np]$ with high probability, where $\delta$ is tiny constant.
7. $T$ is in a ring $R$ centered at $S$, with inner radius $(1 - \delta)np$ and outer radius $(1 + \delta)np$.
8. This ring has

$$\sum_{i=(1-\delta)np}^{(1+\delta)np} \binom{n}{i} \leq 2 \binom{n}{(1 + \delta)np} \leq \alpha = 2 \cdot 2^{n \mathbb{H}((1+\delta)p)}.$$ 

strings in it.
Some intuition...

1. senders sent string $S = s_1 s_2 \ldots s_n$.
2. receiver got string $T = t_1 t_2 \ldots t_n$.
3. $p = \Pr[t_i \neq s_i]$, for all $i$.
4. $U$: Hamming distance between $S$ and $T$: $U = \sum_i [s_i \neq t_i]$.
5. By assumption: $E[U] = pn$, and $U$ is a binomial variable.
6. By Chernoff inequality: $U \in [(1 - \delta)np, (1 + \delta)np]$ with high probability, where $\delta$ is tiny constant.
7. $T$ is in a ring $R$ centered at $S$, with inner radius $(1 - \delta)np$ and outer radius $(1 + \delta)np$.
8. This ring has

$$\sum_{i=(1-\delta)np}^{(1+\delta)np} \binom{n}{i} \leq 2 \binom{n}{(1+\delta)np} \leq \alpha = 2 \cdot 2^{n \mathbb{H}((1+\delta)p)}.$$

strings in it.
Some intuition...

1. senders sent string $S = s_1 s_2 \ldots s_n$.
2. receiver got string $T = t_1 t_2 \ldots t_n$.
3. $p = \Pr[t_i \neq s_i]$, for all $i$.
4. $U$: Hamming distance between $S$ and $T$: $U = \sum_i [s_i \neq t_i]$.
5. By assumption: $E[U] = pn$, and $U$ is a binomial variable.
6. By Chernoff inequality: $U \in [(1 - \delta)np, (1 + \delta)np]$ with high probability, where $\delta$ is tiny constant.
7. $T$ is in a ring $R$ centered at $S$, with inner radius $(1 - \delta)np$ and outer radius $(1 + \delta)np$.
8. This ring has

$$\sum_{i=(1-\delta)np}^{(1+\delta)np} \binom{n}{i} \leq 2 \binom{n}{(1 + \delta)np} \leq \alpha = 2 \cdot 2^{nH((1+\delta)p)}.$$
Some intuition...

1. senders sent string $S = s_1 s_2 \ldots s_n$.
2. receiver got string $T = t_1 t_2 \ldots t_n$.
3. $p = \Pr[t_i \neq s_i]$, for all $i$.
4. $U$: Hamming distance between $S$ and $T$: $U = \sum_i [s_i \neq t_i]$.
5. By assumption: $E[U] = pn$, and $U$ is a binomial variable.
6. By Chernoff inequality: $U \in [(1 - \delta)np, (1 + \delta)np]$ with high probability, where $\delta$ is tiny constant.
7. $T$ is in a ring $R$ centered at $S$, with inner radius $(1 - \delta)np$ and outer radius $(1 + \delta)np$.
8. This ring has

$$\sum_{i=(1-\delta)np}^{(1+\delta)np} \binom{n}{i} \leq 2 \binom{n}{(1+\delta)np} \leq \alpha = 2 \cdot 2^{n \mathbb{H}((1+\delta)p)}.$$

strings in it.
Some intuition...

1. senders sent string $S = s_1 s_2 \ldots s_n$.
2. receiver got string $T = t_1 t_2 \ldots t_n$.
3. $p = \Pr[t_i \neq s_i]$, for all $i$.
4. $U$: Hamming distance between $S$ and $T$: $U = \sum_i [s_i \neq t_i]$.
5. By assumption: $E[U] = pn$, and $U$ is a binomial variable.
6. By Chernoff inequality: $U \in [(1 - \delta)np, (1 + \delta)np]$ with high probability, where $\delta$ is tiny constant.
7. $T$ is in a ring $R$ centered at $S$, with inner radius $(1 - \delta)np$ and outer radius $(1 + \delta)np$.
8. This ring has

$$\sum_{i=(1-\delta)np}^{(1+\delta)np} \binom{n}{i} \leq 2 \binom{n}{(1 + \delta)np} \leq \alpha = 2 \cdot 2^{n \mathbb{H}((1+\delta)p)}.$$
Some intuition...

1. senders sent string $S = s_1s_2 \ldots s_n$.
2. receiver got string $T = t_1t_2 \ldots t_n$.
3. $p = \Pr[t_i \neq s_i]$, for all $i$.
4. $U$: Hamming distance between $S$ and $T$: $U = \sum_i [s_i \neq t_i]$.
5. By assumption: $E[U] = pn$, and $U$ is a binomial variable.
6. By Chernoff inequality: $U \in [(1 - \delta)np, (1 + \delta)np]$ with high probability, where $\delta$ is tiny constant.
7. $T$ is in a ring $R$ centered at $S$, with inner radius $(1 - \delta)np$ and outer radius $(1 + \delta)np$.
8. This ring has $\sum_{i=(1-\delta)np}^{(1+\delta)np} \binom{n}{i} \leq 2 \binom{n}{(1 + \delta)np} \leq \alpha = 2 \cdot 2^{nH((1+\delta)\rho)}$. 

strings in it.
Some intuition...

1. senders sent string $S = s_1 s_2 \ldots s_n$.
2. receiver got string $T = t_1 t_2 \ldots t_n$.
3. $p = \Pr[t_i \neq s_i]$, for all $i$.
4. $U$: Hamming distance between $S$ and $T$: $U = \sum_i [s_i \neq t_i]$.
5. By assumption: $E[U] = pn$, and $U$ is a binomial variable.
6. By Chernoff inequality: $U \in [(1 - \delta)np, (1 + \delta)np]$ with high probability, where $\delta$ is tiny constant.
7. $T$ is in a ring $R$ centered at $S$, with inner radius $(1 - \delta)np$ and outer radius $(1 + \delta)np$.
8. This ring has

$$\sum_{i=(1-\delta)np}^{(1+\delta)np} \binom{n}{i} \leq 2 \binom{n}{(1 + \delta)np} \leq \alpha = 2 \cdot 2^n \mathbb{H}((1+\delta)p).$$

strings in it.
Many rings for many codewords...
Pick as many disjoint rings as possible: \( R_1, \ldots, R_\kappa \).

If every word in the hypercube would be covered...

... use \( 2^n \) codewords \( \implies \kappa \geq \frac{2^n}{|R|} \geq \frac{2^n}{2 \cdot 2^n \mathbb{H}((1+\delta)p)} \approx 2^n(1-\mathbb{H}((1+\delta)p)) \).

Consider all possible strings of length \( k \) such that \( 2^k \leq \kappa \).

Map \( i \)th string in \( \{0,1\}^k \) to the center \( C_i \) of the \( i \)th ring \( R_i \).

If send \( C_i \implies \) receiver gets a string in \( R_i \).

Decoding is easy - find the ring \( R_i \) containing the received string, take its center string \( C_i \), and output the original string it was mapped to.

How many bits?

\[ k = \lfloor \log \kappa \rfloor = n \left( 1 - \mathbb{H}((1 + \delta)p) \right) \approx n(1 - \mathbb{H}(p)) \]
Some more intuition...

1. Pick as many disjoint rings as possible: $R_1, \ldots, R_\kappa$.

2. If every word in the hypercube would be covered...

3. ... use $2^n$ codewords $\implies \kappa \geq \frac{2^n}{|R|} \geq \frac{2^n}{2 \cdot 2^{nH((1+\delta)p)}} \approx 2^n(1-H((1+\delta)p))$.

4. Consider all possible strings of length $k$ such that $2^k \leq \kappa$.

5. Map $i$th string in $\{0, 1\}^k$ to the center $C_i$ of the $i$th ring $R_i$.

6. If send $C_i \implies$ receiver gets a string in $R_i$.

7. Decoding is easy - find the ring $R_i$ containing the received string, take its center string $C_i$, and output the original string it was mapped to.

8. How many bits?

$$k = \lfloor \log \kappa \rfloor = n \left(1 - H((1 + \delta)p)\right) \approx n(1 - H(p)),$$
Some more intuition...

1. Pick as many disjoint rings as possible: \( R_1, \ldots, R_\kappa \).
2. If every word in the hypercube would be covered...

\[ \kappa \geq \frac{2^n}{|R|} \geq \frac{2^n}{2 \cdot 2^n H((1+\delta)p)} \approx 2^n(1-H((1+\delta)p)). \]

3. ... use \( 2^n \) codewords \( \implies \kappa \geq \)

4. Consider all possible strings of length \( k \) such that \( 2^k \leq \kappa \).
5. Map \( i \)th string in \( \{0, 1\}^k \) to the center \( C_i \) of the \( i \)th ring \( R_i \).
6. If send \( C_i \implies \) receiver gets a string in \( R_i \).
7. Decoding is easy - find the ring \( R_i \) containing the received string, take its center string \( C_i \), and output the original string it was mapped to.
8. How many bits?

\[ k = \lfloor \log \kappa \rfloor = n \left( 1 - H\left( (1 + \delta)p \right) \right) \approx n(1 - H(p)). \]
Some more intuition...

1. Pick as many disjoint rings as possible: $R_1, \ldots, R_\kappa$.
2. If every word in the hypercube would be covered...
3. ... use $2^n$ codewords $\implies \kappa \geq \frac{2^n}{|R|} \geq \frac{2^n}{2 \cdot 2^n \mathbb{H}((1+\delta)p)} \approx 2^n(1-\mathbb{H}((1+\delta)p))$.

4. Consider all possible strings of length $k$ such that $2^k \leq \kappa$.
5. Map $i$th string in $\{0, 1\}^k$ to the center $C_i$ of the $i$th ring $R_i$.
6. If send $C_i \implies$ receiver gets a string in $R_i$.
7. Decoding is easy - find the ring $R_i$ containing the received string, take its center string $C_i$, and output the original string it was mapped to.
8. How many bits?

$$k = \lfloor \log \kappa \rfloor = n \left( 1 - \mathbb{H}((1 + \delta)p) \right) \approx n(1 - \mathbb{H}(p)),$$
Some more intuition...

1. Pick as many disjoint rings as possible: $R_1, \ldots, R_\kappa$.
2. If every word in the hypercube would be covered...
3. ... use $2^n$ codewords $\implies \kappa \geq \frac{2^n}{|R|} \geq \frac{2^n}{2 \cdot 2^n H((1+\delta)p)} \approx 2^n (1 - H((1+\delta)p))$.

4. Consider all possible strings of length $k$ such that $2^k \leq \kappa$.
5. Map $i$th string in $\{0, 1\}^k$ to the center $C_i$ of the $i$th ring $R_i$.
6. If send $C_i \implies$ receiver gets a string in $R_i$.
7. Decoding is easy - find the ring $R_i$ containing the received string, take its center string $C_i$, and output the original string it was mapped to.
8. How many bits?

$$k = \lfloor \log \kappa \rfloor = n \left( 1 - H((1 + \delta)p) \right) \approx n(1 - H(p)),$$
Some more intuition...

1. Pick as many disjoint rings as possible: $R_1, \ldots, R_\kappa$.
2. If every word in the hypercube would be covered...
3. ... use $2^n$ codewords $\implies \kappa \geq \frac{2^n}{|R|} \geq \frac{2^n}{2 \cdot 2^{n\mathbb{H}((1+\delta)p)}} \approx 2^n(1-\mathbb{H}((1+\delta)p))$.
4. Consider all possible strings of length $k$ such that $2^k \leq \kappa$.
5. Map $i$th string in $\{0, 1\}^k$ to the center $C_i$ of the $i$th ring $R_i$.
6. If send $C_i \implies$ receiver gets a string in $R_i$.
7. Decoding is easy - find the ring $R_i$ containing the received string, take its center string $C_i$, and output the original string it was mapped to.
8. How many bits?

$$k = \lfloor \log \kappa \rfloor = n\left(1 - \mathbb{H}\left((1 + \delta)p\right)\right) \approx n(1 - \mathbb{H}(p)),$$
Some more intuition...

1. Pick as many disjoint rings as possible: $R_1, \ldots, R_\kappa$.

2. If every word in the hypercube would be covered...

3. ... use $2^n$ codewords \(\implies\) $\kappa \geq \frac{2^n}{|R|} \geq \frac{2^n}{2 \cdot 2^n \mathbb{H}((1+\delta)p)} \approx 2^n(1-\mathbb{H}((1+\delta)p))$.

4. Consider all possible strings of length $k$ such that $2^k \leq \kappa$.

5. Map $i$th string in $\{0, 1\}^k$ to the center $C_i$ of the $i$th ring $R_i$.

6. If send $C_i \implies$ receiver gets a string in $R_i$.

7. Decoding is easy - find the ring $R_i$ containing the received string, take its center string $C_i$, and output the original string it was mapped to.

8. How many bits?

\[ k = \lfloor \log \kappa \rfloor = n \left(1 - \mathbb{H}((1+\delta)p)\right) \approx n(1 - \mathbb{H}(p)), \]
Some more intuition...

1. Pick as many disjoint rings as possible: $R_1, \ldots, R_\kappa$.
2. If every word in the hypercube would be covered...
3. ... use $2^n$ codewords $\implies \kappa \geq \frac{2^n}{|R|} \geq \frac{2^n}{2 \cdot 2^n \mathbb{H}((1+\delta)p)} \approx 2^n(1-H((1+\delta)p))$.
4. Consider all possible strings of length $k$ such that $2^k \leq \kappa$.
5. Map $i$th string in $\{0,1\}^k$ to the center $C_i$ of the $i$th ring $R_i$.
6. If send $C_i \implies$ receiver gets a string in $R_i$.
7. Decoding is easy - find the ring $R_i$ containing the received string, take its center string $C_i$, and output the original string it was mapped to.
8. How many bits?

$$k = \lfloor \log \kappa \rfloor = n \left(1 - \mathbb{H}\left((1 + \delta)p\right)\right) \approx n(1 - \mathbb{H}(p)),$$
Some more intuition...

1. Pick as many disjoint rings as possible: $R_1, \ldots, R_\kappa$.
2. If every word in the hypercube would be covered...
3. ... use $2^n$ codewords $\implies \kappa \geq$

$$\kappa \geq \frac{2^n}{|R|} \geq \frac{2^n}{2 \cdot 2^{n \mathbb{H}((1+\delta)p)}} \approx 2^n(1-\mathbb{H}((1+\delta)p)).$$

4. Consider all possible strings of length $k$ such that $2^k \leq \kappa$.
5. Map $i$th string in $\{0, 1\}^k$ to the center $C_i$ of the $i$th ring $R_i$.
6. If send $C_i \implies$ receiver gets a string in $R_i$.
7. Decoding is easy - find the ring $R_i$ containing the received string, take its center string $C_i$, and output the original string it was mapped to.
8. How many bits?

$$k = \lfloor \log \kappa \rfloor = n \left(1 - \mathbb{H}((1 + \delta)p)\right) \approx n(1 - \mathbb{H}(p)),$$
What is wrong with the above?

1. Can not find such a large set of disjoint rings.
2. Reason is that when you pack rings (or balls) you are going to have wasted spaces around.
3. Overcome this: allow rings to overlap somewhat.
4. Makes things considerably more involved.
5. Details in class notes.
What is wrong with the above?

1. Can not find such a large set of disjoint rings.
2. Reason is that when you pack rings (or balls) you are going to have wasted spaces around.
3. Overcome this: allow rings to overlap somewhat.
4. Makes things considerably more involved.
5. Details in class notes.
What is wrong with the above?

1. Can not find such a large set of disjoint rings.
2. Reason is that when you pack rings (or balls) you are going to have wasted spaces around.
3. Overcome this: allow rings to overlap somewhat.
4. Makes things considerably more involved.
5. Details in class notes.
What is wrong with the above?

1. Can not find such a large set of disjoint rings.
2. Reason is that when you pack rings (or balls) you are going to have wasted spaces around.
3. Overcome this: allow rings to overlap somewhat.
4. Makes things considerably more involved.
5. Details in class notes.
What is wrong with the above?

1. Can not find such a large set of disjoint rings.
2. Reason is that when you pack rings (or balls) you are going to have wasted spaces around.
3. Overcome this: allow rings to overlap somewhat.
4. Makes things considerably more involved.
5. Details in class notes.