04

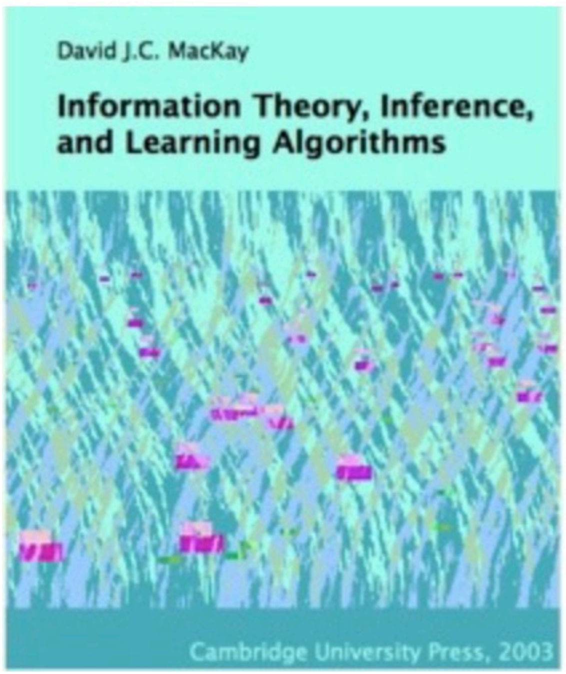
Data Compression

Notice

- **Author**
  - **João Moura Pires (jmp@fct.unl.pt)**

Bibliography

- **Many examples are extracted and adapted from:**



Information Theory, Inference, and Learning Algorithms
David J.C. MacKay
2005, Version 7.2

- **And some slides were based on Iain Murray course**
  - **http://www.inf.ed.ac.uk/teaching/courses/it/2014/**

Table of Contents

Data Compression

Data Compression and Shanon Information content

Data Compression and Shanon Information content

- The **Shannon information content** of an outcome *x* is a natural measure of its **information content**
  - **Improbable outcomes** do convey **more information** than probable outcomes

- **Information content of a source** by considering **how many bits are needed to describe the outcome of an experiment**

Data Compression and Shanon Information content

- The **Shannon information content** of an outcome *x* is a natural measure of its **information content**
  - **Improbable outcomes** do convey **more information** than probable outcomes

- **Information content of a source** by considering **how many bits are needed to describe the outcome of an experiment**
  - If we can show that we can compress data from a particular source into a file of **L bits per source symbol** and recover the data reliably,

Data Compression and Shanon Information content

- The **Shannon information content** of an outcome *x* is a natural measure of its **information content**
  - **Improbable outcomes** do convey **more information** than probable outcomes

- **Information content of a source** by considering **how many bits are needed to describe the outcome of an experiment**
  - If we can show that we can compress data from a particular source into a file of **L bits per source symbol** and recover the data reliably,
  - then we will say that the **average information content of that source is at most L bits per symbol**.

Simple data compression methods for $|A_X|$

- One way of measuring the **information content of a random variable** is simply to **count the number of possible outcomes**, $|A_X|$.

FACULDADE DE CIÊNCIAS E TECNOLOGIA UNIVERSIDADE NOVA DE LISBOA

Simple data compression methods for $|A_X|$

- One way of measuring the **information content of a random variable** is simply to **count the number of possible outcomes**, $|A_X|$.
- If we gave a **binary name to each outcome**, the **length of each name would** be $\log_2 |A_X|$ bits, if $|A_X|$ happened to be a power of 2.

Simple data compression methods for $|A_X|$

- One way of measuring the **information content of a random variable** is simply to **count the number of possible outcomes**, $|A_X|$.
- If we gave a **binary name to each outcome**, the **length of each name would** be $\log_2 |A_X|$ bits, if $|A_X|$ happened to be a power of 2.
- The **raw bit content** of X is

$$H_0(X) = \log_2 |A_X|$$

Simple data compression methods for $|A_X|$

- One way of measuring the **information content of a random variable** is simply to **count the number of possible outcomes**, $|A_X|$.
- If we gave a **binary name to each outcome**, the **length of each name would** be $\log_2 |A_X|$ bits, if $|A_X|$ happened to be a power of 2.
- The **raw bit content** of X is

$$H_0(X) = \log_2 |A_X|$$

- It is an additive quantity: the raw bit content of an ordered pair x, y, having $|A_X|\,|A_Y|$ possible outcomes, satisfies:

$$H_0(X,Y) = H_0(X) + H_0(Y)$$

Simple data compression methods for $|A_X|$

- One way of measuring the **information content of a random variable** is simply to **count the number of possible outcomes**, $|A_X|$.
- If we gave a **binary name to each outcome**, the **length of each name would** be $\log_2 |A_X|$ bits, if $|A_X|$ happened to be a power of 2.
- The **raw bit content** of X is

$$H_0(X) = \log_2 |A_X|$$

- It is an additive quantity: the raw bit content of an ordered pair x, y, having $|A_X|\,|A_Y|$ possible outcomes, satisfies:

$$H_0(X,Y) = H_0(X) + H_0(Y)$$

- Does not include any probabilistic element, and the encoding rule does not 'compress'.

Compress all filles?

- Could there be:
  - a compressor that maps an outcome x to a binary code c(x),
  - a decompressor that maps c back to x,
  - **such that every possible outcome is compressed into a binary code of length shorter than $H_0(X)$ bits?**

Compress all filles?

- Could there be:
  - a compressor that maps an outcome x to a binary code c(x),
  - a decompressor that maps c back to x,
  - **such that every possible outcome is compressed into a binary code of length shorter than $H_0(X)$ bits?**

- No !! It is impossible to make a reversible compression program that **reduces the size of all files**

Ways for compressing files

Ways for compressing files

- A *lossy* **compressor** compresses some files, but **maps some files to the same encoding**.
  - We'll denote by **δ** the probability that the source string is one of the confusable files, so a lossy compressor has a probability **δ** of failure.
  - If **δ** can be made very small then a lossy compressor may be practically useful. (images, videos, etc)

Ways for compressing files

- A *lossy* **compressor** compresses some files, but **maps some files to the same encoding**.
  - We'll denote by **δ** the probability that the source string is one of the confusable files, so a lossy compressor has a probability **δ** of failure.
  - If **δ** can be made very small then a lossy compressor may be practically useful. (images, videos, etc)
- A *lossless* **compressor** maps **all files to different encodings**
  - if it **shortens some files**, it necessarily **makes others longer**.
  - We try to design the compressor so that the **probability that a file is lengthened is very small**, and the probability that it is shortened is large.

Information content in terms of lossy compression

Take into account the **probabilities** of the different outcomes

- Imagine comparing the information contents of two text files

  A. one in which all 128 ASCII characters are used with **equal probability**

  B. one in which the characters are used with their **frequencies in English text**

Take into account the **probabilities** of the different outcomes

- Imagine comparing the information contents of two text files

  A. one in which all 128 ASCII characters are used with **equal probability**

  B. one in which the characters are used with their **frequencies in English text**

- Can we define a measure of information content that distinguishes between these two files?
  - The case B. contains less information per character because it is more predictable

Take into account the **probabilities** of the different outcomes

- Imagine comparing the information contents of two text files
  - A. one in which all 128 ASCII characters are used with **equal probability**
  - B. one in which the characters are used with their **frequencies in English text**
- Can we define a measure of information content that distinguishes between these two files?
  - The case B. contains less information per character because it is more predictable
- How to use this knowledge?
  - For instance **just remove the less probable symbols to get a smaller alphabet**
  - For instance, guessing that the most infrequent characters { !, @, #, %, ^, *, ~, <, >, /, \, _, {, },

    [,], | } won't occur ! — Reducing the alphabet by seventeen.
  - **δ** - is the probability that there will be no name for an outcome *x*.

FACULDADE DE CIÊNCIAS E TECNOLOGIA UNIVERSIDADE NOVA DE LISBOA

Take into account the **probabilities** of the different outcomes

- Example:

$$\mathcal{A}_X = \{\, \texttt{a}, \texttt{b}, \texttt{c}, \texttt{d}, \texttt{e}, \texttt{f}, \texttt{g}, \texttt{h} \,\},$$
$$\mathcal{P}_X = \{\, \tfrac{1}{4}, \tfrac{1}{4}, \tfrac{1}{4}, \tfrac{3}{16}, \tfrac{1}{64}, \tfrac{1}{64}, \tfrac{1}{64}, \tfrac{1}{64} \,\}.$$

$$H_0(X) = \log_2 |A_X| = \log_2 8 = 3 bits$$

Take into account the **probabilities** of the different outcomes

- Example:

$$\mathcal{A}_X = \{\,\texttt{a},\,\texttt{b},\,\texttt{c},\,\texttt{d},\,\texttt{e},\,\texttt{f},\,\texttt{g},\,\texttt{h}\,\},$$
$$\mathcal{P}_X = \{\,\tfrac{1}{4},\tfrac{1}{4},\tfrac{1}{4},\tfrac{3}{16},\tfrac{1}{64},\tfrac{1}{64},\tfrac{1}{64},\tfrac{1}{64}\,\}.$$

$$H_0(X) = \log_2 |A_X| = \log_2 8 = 3 bits$$

- But $P(x \in \{a, b, c, d\}) = 15/16$.

Take into account the **probabilities** of the different outcomes

- Example:

$$\mathcal{A}_X = \{ \texttt{a}, \texttt{b}, \texttt{c}, \texttt{d}, \texttt{e}, \texttt{f}, \texttt{g}, \texttt{h} \},$$
$$\mathcal{P}_X = \{ \tfrac{1}{4}, \tfrac{1}{4}, \tfrac{1}{4}, \tfrac{3}{16}, \tfrac{1}{64}, \tfrac{1}{64}, \tfrac{1}{64}, \tfrac{1}{64} \}.$$

$$H_0(X) = \log_2 |A_X| = \log_2 8 = 3 bits$$

- But $P(x \in \{a, b, c, d\}) = 15/16$.
- So if we accept a risk $\delta = 1/16$ of not having a symbol for x, we can consider codes only for each in {a, b, c, d} and so only requiring 2 bits.

$\delta = 0$		$\delta = 1/16$	
x	$c(x)$	x	$c(x)$
a	000	a	00
b	001	b	01
c	010	c	10
d	011	d	11
e	100	e	–
f	101	f	–
g	110	g	–
h	111	h	–

Smallest δ-sufficient subset

- **The smallest δ-sufficient subset**.
  - S_δ is the smallest subset of A_X satisfying

$$P(x \in S_\partial) \geq 1 - \partial$$

Smallest δ-sufficient subset

- **The smallest δ-sufficient subset**.
  - S_δ is the smallest subset of A_X satisfying

$$P(x \in S_\partial) \geq 1 - \partial$$

- The subset S_δ can be constructed by **ranking the elements of A_X in order of decreasing probability** and adding successive elements starting from the most probable elements until the total probability is ≥ (1−δ).

Smallest δ-sufficient subset

- **The smallest δ-sufficient subset**.
  - S_δ is the smallest subset of A_X satisfying

$$P(x \in S_\partial) \geq 1 - \partial$$

- The subset S_δ can be constructed by **ranking the elements of A_X in order of decreasing probability** and adding successive elements starting from the most probable elements until the total probability is ≥ (1–δ).

- We can make a **data compression code** by **assigning a binary name to each element of the smallest sufficient subset**

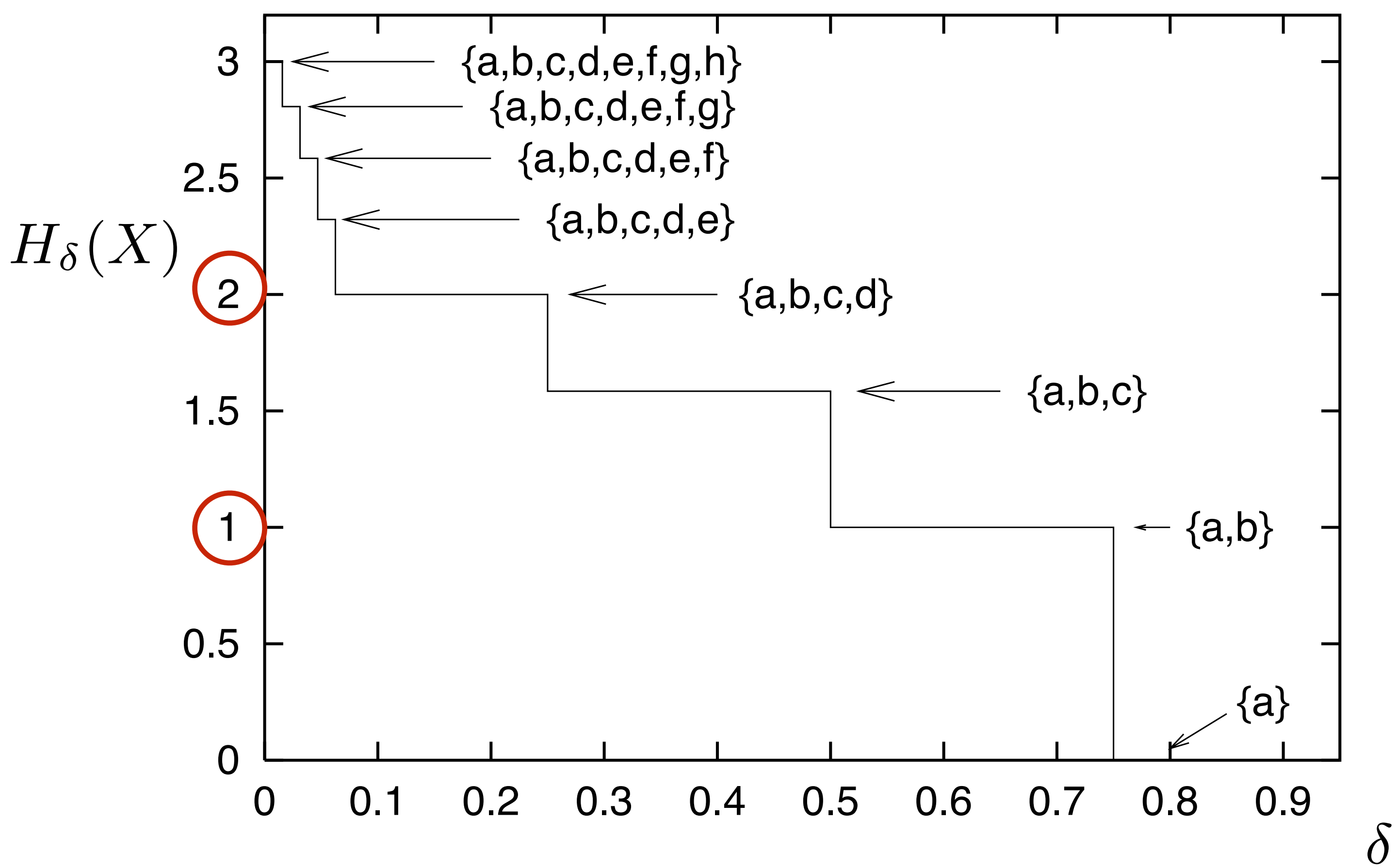
Essential bit content of X

- **The essential bit content of X** is:

$$H_\partial(X) = \log_2 |S_\partial|$$

- Note that $H_0(X)$ is the special case of $H_\partial(X)$ with $\partial = 0$ (if $P(x) > 0$ for all $x \in A_X$).

$$\mathcal{A}_X = \{ \texttt{a}, \texttt{b}, \texttt{c}, \texttt{d}, \texttt{e}, \texttt{f}, \texttt{g}, \texttt{h} \},$$
$$\mathcal{P}_X = \{ \tfrac{1}{4}, \tfrac{1}{4}, \tfrac{1}{4}, \tfrac{3}{16}, \tfrac{1}{64}, \tfrac{1}{64}, \tfrac{1}{64}, \tfrac{1}{64} \}.$$



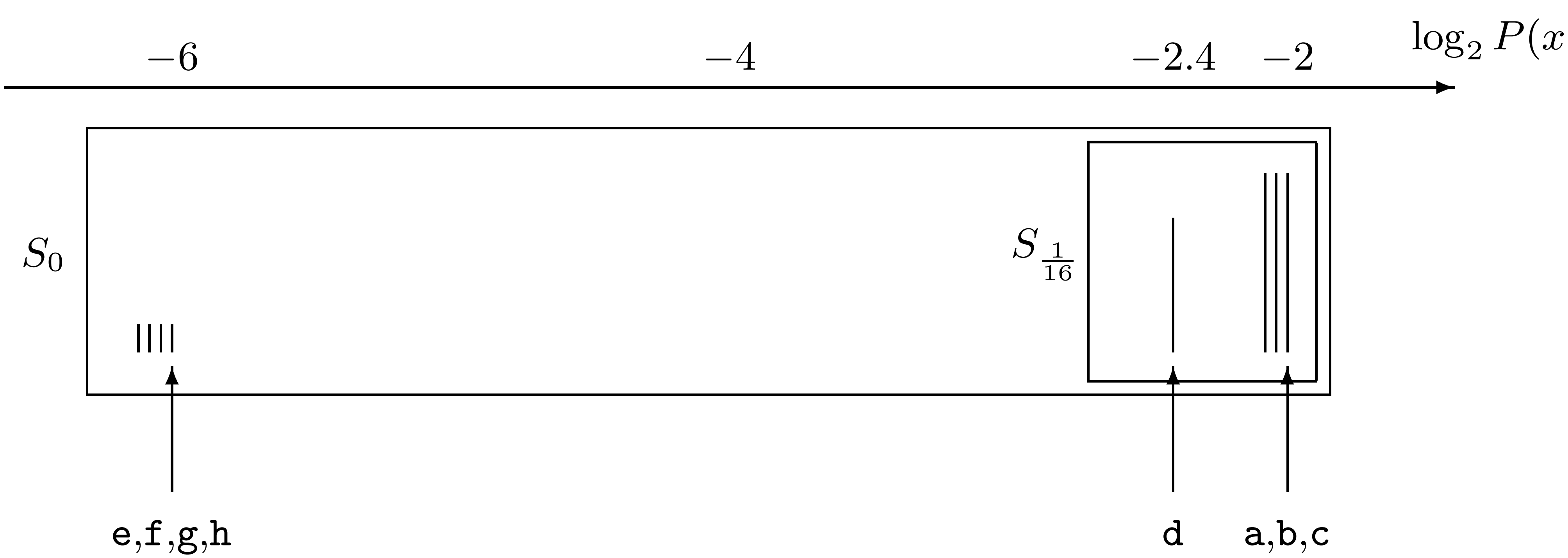
Essential bit content of X

- **The essential bit content of X** is:

$$H_\partial(X) = \log_2 |S_\partial|$$

- Note that $H_0(X)$ is the special case of $H_\partial(X)$ with $\partial = 0$ (if $P(x) > 0$ for all $x \in A_X$).

$$\mathcal{A}_X = \{\, \mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, \mathtt{e}, \mathtt{f}, \mathtt{g}, \mathtt{h} \,\},$$
$$\mathcal{P}_X = \{\, \tfrac{1}{4}, \tfrac{1}{4}, \tfrac{1}{4}, \tfrac{3}{16}, \tfrac{1}{64}, \tfrac{1}{64}, \tfrac{1}{64}, \tfrac{1}{64} \,\}.$$



Extended ensembles

- What if we **compress *blocks* of symbols** from a source?
- Let the outcome $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)$ is a string of N independent identically distributed random variables from a single ensemble X. X^N is the ensemble $(X_1, X_2, \ldots, X_N)$

Extended ensembles

- What if we **compress *blocks* of symbols** from a source?
- Let the outcome $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)$ is a string of N independent identically distributed random variables from a single ensemble X. X^N is the ensemble $(X_1, X_2, \ldots, X_N)$
- The Entropy is additive for independent variables: $H(X^N) = N\,H(X)$

Extended ensembles

- What if we **compress *blocks* of symbols** from a source?
- Let the outcome $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)$ is a string of N independent identically distributed random variables from a single ensemble X. X^N is the ensemble $(X_1, X_2, \ldots, X_N)$
- The Entropy is additive for independent variables: $H(X^N) = N\, H(X)$
- Example:
  - N flips of a bent coin, $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)$, where $x_i \in \{0, 1\}$, with $p_0 = 0.9$ and $p_1 = 0.1$
  - The most probable sequences are those with most 0s.
  - If $r(\boldsymbol{x})$ is the number of 1s in $\boldsymbol{x}$ then

$$P(\boldsymbol{x}) = p_0^{\,N-r(\boldsymbol{x})}\, p_1^{\,r(\boldsymbol{x})}$$

Extended ensembles

- N flips of a bent coin, $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)$, where $x_i \in \{0, 1\}$, with **$p_0 = 0.9$** and **$p_1 = 0.1$**
- The **most probable sequences** are those **with most 0s**.
- If $r(\boldsymbol{x})$ is the number of 1s in $\boldsymbol{x}$ then $P(\boldsymbol{x}) = p_0^{N-r(\boldsymbol{x})} p_1^{r(\boldsymbol{x})}$
- To evaluate $H_\partial(X^N)$ we must find the smallest sufficient subset S_δ
  - S_δ is the subset that contains all sequences $\boldsymbol{x}$ with $r(\boldsymbol{x}) = 0, 1, \ldots, r_{max}(\partial) - 1$ and some sequences with $r(\boldsymbol{x}) = r_{max}(\partial)$.

Extended ensembles - Example for *N* = 4

- To evaluate $H_{\partial}(X^N)$ we must find the smallest sufficient subset S_δ
  - S_δ is the subset that contains all sequences $\boldsymbol{x}$ with $r(\boldsymbol{x}) = 0, 1, \ldots, r_{max}(\partial) - 1$ and some sequences with $r(\boldsymbol{x}) = r_{max}(\partial)$.

Extended ensembles - Example for *N* = 4

- To evaluate $H_\partial(X^N)$ we must find the smallest sufficient subset S_δ
  - S_δ is the subset that contains all sequences $\boldsymbol{x}$ with $r(\boldsymbol{x}) = 0, 1, \ldots, r_{max}(\partial) - 1$ and some sequences with $r(\boldsymbol{x}) = r_{max}(\partial)$.

$r(\boldsymbol{x})$
0
1
2
3
4

Extended ensembles - Example for *N* = 4

- To evaluate $H_\partial(X^N)$ we must find the smallest sufficient subset S_δ
  - S_δ is the subset that contains all sequences $\boldsymbol{x}$ with $r(\boldsymbol{x}) = 0, 1, \ldots, r_{max}(\partial) - 1$ and some sequences with $r(\boldsymbol{x}) = r_{max}(\partial)$.

$$P(\boldsymbol{x}) = p_0^{N-r(\boldsymbol{x})} p_1^{r(\boldsymbol{x})}$$

$r(\boldsymbol{x})$
0
1
2
3
4

Extended ensembles - Example for ***N* = 4**

- To evaluate $H_\partial(X^N)$ we must find the smallest sufficient subset S_δ
  - S_δ is the subset that contains all sequences $\boldsymbol{x}$ with $r(\boldsymbol{x}) = 0, 1, \ldots, r_{\max}(\partial) - 1$ and some sequences with $r(\boldsymbol{x}) = r_{\max}(\partial)$.

$$P(\boldsymbol{x}) = p_0^{N-r(\boldsymbol{x})} p_1^{r(\boldsymbol{x})}$$

r(**x**)	*P*(**x**)
0	0,6561
1	0,0729
2	0,0081
3	0,0009
4	1E-04

Extended ensembles - Example for *N* = 4

- To evaluate $H_\partial(X^N)$ we must find the smallest sufficient subset S_δ
  - S_δ is the subset that contains all sequences $\boldsymbol{x}$ with $r(\boldsymbol{x}) = 0, 1, \ldots, r_{max}(\partial) - 1$ and some sequences with $r(\boldsymbol{x}) = r_{max}(\partial)$.

$$P(\boldsymbol{x}) = p_0^{N-r(\boldsymbol{x})} p_1^{r(\boldsymbol{x})}$$

$r(\mathbf{x})$	$P(\mathbf{x})$	$\log_2 P(\mathbf{x})$
0	0,6561	-0,6
1	0,0729	-3,8
2	0,0081	-6,9
3	0,0009	-10,1
4	1E-04	-13,3

Extended ensembles - Example for *N* = 4

- To evaluate $H_\partial(X^N)$ we must find the smallest sufficient subset S_δ
  - S_δ is the subset that contains all sequences $\boldsymbol{x}$ with $r(\boldsymbol{x}) = 0, 1, \ldots, r_{max}(\partial) - 1$ and some sequences with $r(\boldsymbol{x}) = r_{max}(\partial)$.

$$P(\boldsymbol{x}) = p_0^{\,N-r(\boldsymbol{x})} p_1^{\,r(\boldsymbol{x})}$$

$r(\mathbf{x})$	$P(\mathbf{x})$	$\log_2 P(\mathbf{x})$	$C(N, r)$
0	0,6561	-0,6	1
1	0,0729	-3,8	4
2	0,0081	-6,9	6
3	0,0009	-10,1	4
4	1E-04	-13,3	1

Extended ensembles - Example for ***N* = 4**

- To evaluate $H_\partial(X^N)$ we must find the smallest sufficient subset S_δ

n
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

Extended ensembles - Example for ***N* = 4**

- To evaluate $H_\partial(X^N)$ we must find the smallest sufficient subset S_δ

n	x_1	x_2	x_3	x_4
1	0	0	0	0
2	0	0	0	1
3	0	0	1	0
4	0	1	0	0
5	1	0	0	0
6	0	0	1	1
7	0	1	0	1
8	0	1	1	0
9	1	0	0	1
10	1	0	1	0
11	1	1	0	0
12	0	1	1	1
13	1	0	1	1
14	1	1	0	1
15	1	1	1	0
16	1	1	1	1

Extended ensembles - Example for *N* = 4

- To evaluate $H_\partial(X^N)$ we must find the smallest sufficient subset S_δ

$r(x)$ is the number of 1s in x

n	x_1	x_2	x_3	x_4	$r(\mathbf{x})$
1	0	0	0	0	0
2	0	0	0	1	1
3	0	0	1	0	1
4	0	1	0	0	1
5	1	0	0	0	1
6	0	0	1	1	2
7	0	1	0	1	2
8	0	1	1	0	2
9	1	0	0	1	2
10	1	0	1	0	2
11	1	1	0	0	2
12	0	1	1	1	3
13	1	0	1	1	3
14	1	1	0	1	3
15	1	1	1	0	3
16	1	1	1	1	4

Extended ensembles - Example for *N* = 4

- To evaluate $H_\partial(X^N)$ we must find the smallest sufficient subset S_δ

n	x_1	x_2	x_3	x_4	$r(\mathbf{x})$
1	0	0	0	0	0
2	0	0	0	1	1
3	0	0	1	0	1
4	0	1	0	0	1
5	1	0	0	0	1
6	0	0	1	1	2
7	0	1	0	1	2
8	0	1	1	0	2
9	1	0	0	1	2
10	1	0	1	0	2
11	1	1	0	0	2
12	0	1	1	1	3
13	1	0	1	1	3
14	1	1	0	1	3
15	1	1	1	0	3
16	1	1	1	1	4

$r(x)$ is the number of 1s in x

$$P(\mathbf{x}) = p_0^{N-r(\mathbf{x})} p_1^{r(\mathbf{x})}$$

$p_0 = 0.9$ and $p_1 = 0.1$

Extended ensembles - Example for *N* = 4

- To evaluate $H_\partial(X^N)$ we must find the smallest sufficient subset S_δ

n	x_1	x_2	x_3	x_4	$r(\boldsymbol{x})$	$P(\boldsymbol{x})$
1	0	0	0	0	0	0,6561
2	0	0	0	1	1	0,0729
3	0	0	1	0	1	0,0729
4	0	1	0	0	1	0,0729
5	1	0	0	0	1	0,0729
6	0	0	1	1	2	0,0081
7	0	1	0	1	2	0,0081
8	0	1	1	0	2	0,0081
9	1	0	0	1	2	0,0081
10	1	0	1	0	2	0,0081
11	1	1	0	0	2	0,0081
12	0	1	1	1	3	0,0009
13	1	0	1	1	3	0,0009
14	1	1	0	1	3	0,0009
15	1	1	1	0	3	0,0009
16	1	1	1	1	4	0,0001

$r(x)$ is the number of 1s in x

$$P(\boldsymbol{x}) = p_0^{N-r(\boldsymbol{x})} p_1^{r(\boldsymbol{x})}$$

$p_0 = 0.9$ and $p_1 = 0.1$

Extended ensembles - Example for *N* = 4

- To evaluate $H_\partial(X^N)$ we must find the smallest sufficient subset S_δ

n	x_1	x_2	x_3	x_4	$r(\boldsymbol{x})$	$P(\boldsymbol{x})$	$\log_2 P(\boldsymbol{x})$
1	0	0	0	0	0	0,6561	-0,6
2	0	0	0	1	1	0,0729	-3,8
3	0	0	1	0	1	0,0729	-3,8
4	0	1	0	0	1	0,0729	-3,8
5	1	0	0	0	1	0,0729	-3,8
6	0	0	1	1	2	0,0081	-6,9
7	0	1	0	1	2	0,0081	-6,9
8	0	1	1	0	2	0,0081	-6,9
9	1	0	0	1	2	0,0081	-6,9
10	1	0	1	0	2	0,0081	-6,9
11	1	1	0	0	2	0,0081	-6,9
12	0	1	1	1	3	0,0009	-10,1
13	1	0	1	1	3	0,0009	-10,1
14	1	1	0	1	3	0,0009	-10,1
15	1	1	1	0	3	0,0009	-10,1
16	1	1	1	1	4	0,0001	-13,3

$r(x)$ is the number of 1s in x

$$P(\boldsymbol{x}) = p_0^{N-r(\boldsymbol{x})} p_1^{r(\boldsymbol{x})}$$

$p_0 = 0.9$ and $p_1 = 0.1$

Extended ensembles - Example for ***N* = 4**

- To evaluate $H_\partial(X^N)$ we must find the smallest sufficient subset S_δ

n	x_1	x_2	x_3	x_4	$r(\boldsymbol{x})$	$P(\boldsymbol{x})$	$\log_2 P(\boldsymbol{x})$
1	0	0	0	0	0	0,6561	-0,6
2	0	0	0	1	1	0,0729	-3,8
3	0	0	1	0	1	0,0729	-3,8
4	0	1	0	0	1	0,0729	-3,8
5	1	0	0	0	1	0,0729	-3,8
6	0	0	1	1	2	0,0081	-6,9
7	0	1	0	1	2	0,0081	-6,9
8	0	1	1	0	2	0,0081	-6,9
9	1	0	0	1	2	0,0081	-6,9
10	1	0	1	0	2	0,0081	-6,9
11	1	1	0	0	2	0,0081	-6,9
12	0	1	1	1	3	0,0009	-10,1
13	1	0	1	1	3	0,0009	-10,1
14	1	1	0	1	3	0,0009	-10,1
15	1	1	1	0	3	0,0009	-10,1
16	1	1	1	1	4	0,0001	-13,3

$r(x)$ is the number of 1s in x

$$P(\boldsymbol{x}) = p_0^{N-r(\boldsymbol{x})} p_1^{r(\boldsymbol{x})}$$

$p_0 = 0.9$ and $p_1 = 0.1$

$$H_\partial(X) = \log_2 |S_\partial|$$

Extended ensembles - Example for *N* = 4

- To evaluate $H_\partial(X^N)$ we must find the smallest sufficient subset S_δ

n	x_1	x_2	x_3	x_4	$r(\mathbf{x})$	$P(\mathbf{x})$	$\log_2 P(\mathbf{x})$	$H_\partial(X)$
1	0	0	0	0	0	0,6561	-0,6	0,000
2	0	0	0	1	1	0,0729	-3,8	1,000
3	0	0	1	0	1	0,0729	-3,8	1,585
4	0	1	0	0	1	0,0729	-3,8	2,000
5	1	0	0	0	1	0,0729	-3,8	2,322
6	0	0	1	1	2	0,0081	-6,9	2,585
7	0	1	0	1	2	0,0081	-6,9	2,807
8	0	1	1	0	2	0,0081	-6,9	3,000
9	1	0	0	1	2	0,0081	-6,9	3,170
10	1	0	1	0	2	0,0081	-6,9	3,322
11	1	1	0	0	2	0,0081	-6,9	3,459
12	0	1	1	1	3	0,0009	-10,1	3,585
13	1	0	1	1	3	0,0009	-10,1	3,700
14	1	1	0	1	3	0,0009	-10,1	3,807
15	1	1	1	0	3	0,0009	-10,1	3,907
16	1	1	1	1	4	0,0001	-13,3	4,000

$r(x)$ is the number of 1s in x

$$P(\mathbf{x}) = p_0^{\,N-r(\mathbf{x})} p_1^{\,r(\mathbf{x})}$$

$p_0 = 0.9$ and $p_1 = 0.1$

$$H_\partial(X) = \log_2 |S_\partial|$$

Extended ensembles - Example for $N = 4$

- To evaluate $H_\partial(X^N)$ we must find the smallest sufficient subset S_δ

n	x_1	x_2	x_3	x_4	$r(\mathbf{x})$	$P(\mathbf{x})$	$\log_2 P(\mathbf{x})$	$H_\partial(X)$
1	0	0	0	0	0	0,6561	-0,6	0,000
2	0	0	0	1	1	0,0729	-3,8	1,000
3	0	0	1	0	1	0,0729	-3,8	1,585
4	0	1	0	0	1	0,0729	-3,8	2,000
5	1	0	0	0	1	0,0729	-3,8	2,322
6	0	0	1	1	2	0,0081	-6,9	2,585
7	0	1	0	1	2	0,0081	-6,9	2,807
8	0	1	1	0	2	0,0081	-6,9	3,000
9	1	0	0	1	2	0,0081	-6,9	3,170
10	1	0	1	0	2	0,0081	-6,9	3,322
11	1	1	0	0	2	0,0081	-6,9	3,459
12	0	1	1	1	3	0,0009	-10,1	3,585
13	1	0	1	1	3	0,0009	-10,1	3,700
14	1	1	0	1	3	0,0009	-10,1	3,807
15	1	1	1	0	3	0,0009	-10,1	3,907
16	1	1	1	1	4	0,0001	-13,3	4,000

$r(x)$ is the number of 1s in x

$$P(\mathbf{x}) = p_0^{N-r(\mathbf{x})} p_1^{r(\mathbf{x})}$$

$p_0 = 0.9$ and $p_1 = 0.1$

$$H_\partial(X) = \log_2 |S_\partial|$$

$$P(x \in S_\partial) \geq 1 - \partial$$

Extended ensembles - Example for *N* = 4

- To evaluate $H_\partial(X^N)$ we must find the smallest sufficient subset S_δ

n	x_1	x_2	x_3	x_4	$r(\mathbf{x})$	$P(\mathbf{x})$	$\log_2 P(\mathbf{x})$	$H_\partial(X)$	$P(\mathbf{x} \notin S_\partial)$
1	0	0	0	0	0	0,6561	-0,6	0,000	1,000
2	0	0	0	1	1	0,0729	-3,8	1,000	0,344
3	0	0	1	0	1	0,0729	-3,8	1,585	0,271
4	0	1	0	0	1	0,0729	-3,8	2,000	0,198
5	1	0	0	0	1	0,0729	-3,8	2,322	0,125
6	0	0	1	1	2	0,0081	-6,9	2,585	0,052
7	0	1	0	1	2	0,0081	-6,9	2,807	0,044
8	0	1	1	0	2	0,0081	-6,9	3,000	0,036
9	1	0	0	1	2	0,0081	-6,9	3,170	0,028
10	1	0	1	0	2	0,0081	-6,9	3,322	0,020
11	1	1	0	0	2	0,0081	-6,9	3,459	0,012
12	0	1	1	1	3	0,0009	-10,1	3,585	0,004
13	1	0	1	1	3	0,0009	-10,1	3,700	0,003
14	1	1	0	1	3	0,0009	-10,1	3,807	0,002
15	1	1	1	0	3	0,0009	-10,1	3,907	0,001
16	1	1	1	1	4	0,0001	-13,3	4,000	0,000

r*(*x*) is the number of 1s in *x

$$P(\mathbf{x}) = p_0^{\,N-r(\mathbf{x})} p_1^{\,r(\mathbf{x})}$$

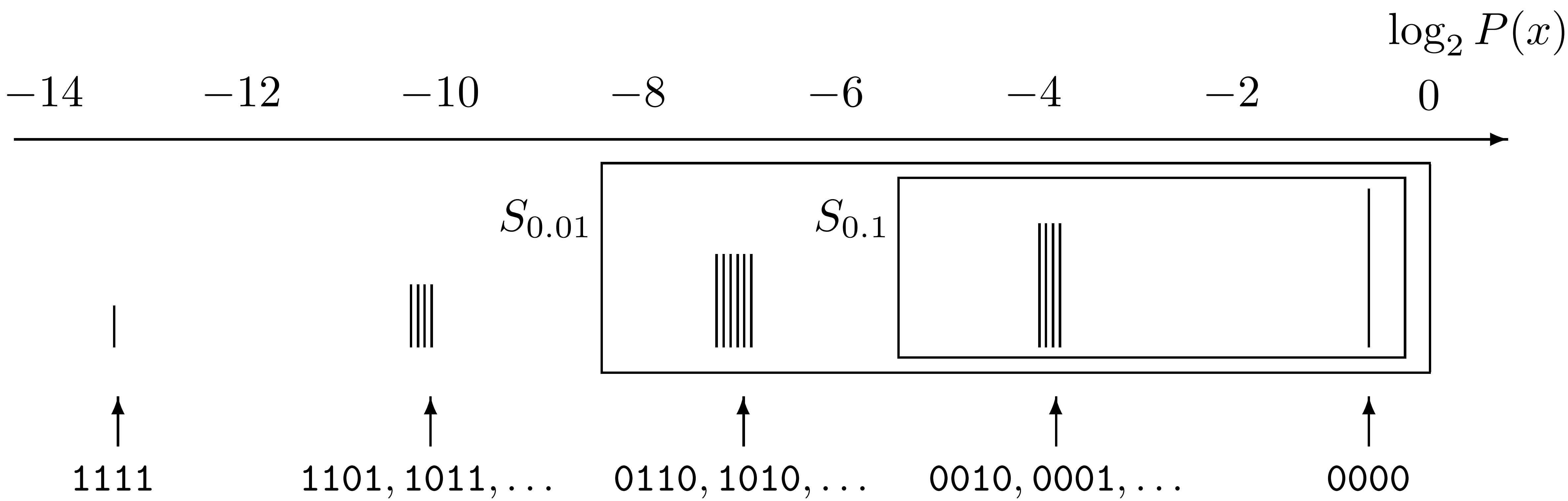
$p_0 = 0.9$ and $p_1 = 0.1$

$$H_\partial(X) = \log_2 |S_\partial|$$

$$P(x \in S_\partial) \geq 1 - \partial$$

Extended ensembles - Example for $N = 4$

- To evaluate $H_\partial(X^N)$ we must find the smallest sufficient subset S_δ
  - S_δ is the subset that contains all sequences $\boldsymbol{x}$ with $r(\boldsymbol{x}) = 0, 1, \ldots, r_{max}(\partial) - 1$ and some sequences with $r(\boldsymbol{x}) = r_{max}(\partial)$.



Extended ensembles - Example for $N = 4$

- To evaluate $H_\partial(X^N)$ we must find the smallest sufficient subset S_δ

	n	x_1	x_2	x_3	x_4	$r(\mathbf{x})$	$P(\mathbf{x})$	$\log_2 P(\mathbf{x})$	$H_\partial(X)$	$P(\mathbf{x} \notin S_\partial)$
1 bit	1	0	0	0	0	0	0,6561	-0,6	0,000	1,00000
	2	0	0	0	1	1	0,0729	-3,8	1,000	**0,34390**
2 bit	3	0	0	1	0	1	0,0729	-3,8	1,585	0,27100
	4	0	1	0	0	1	0,0729	-3,8	2,000	**0,19810**
3 bit	5	1	0	0	0	1	0,0729	-3,8	2,322	0,12520
	6	0	0	1	1	2	0,0081	-6,9	2,585	0,05230
	7	0	1	0	1	2	0,0081	-6,9	2,807	0,04420
	8	0	1	1	0	2	0,0081	-6,9	3,000	**0,03610**
4 bit	9	1	0	0	1	2	0,0081	-6,9	3,170	0,02800
	10	1	0	1	0	2	0,0081	-6,9	3,322	0,01990
	11	1	1	0	0	2	0,0081	-6,9	3,459	0,01180
	12	0	1	1	1	3	0,0009	-10,1	3,585	0,00370
	13	1	0	1	1	3	0,0009	-10,1	3,700	0,00280
	14	1	1	0	1	3	0,0009	-10,1	3,807	0,00190
	15	1	1	1	0	3	0,0009	-10,1	3,907	0,00100
	16	1	1	1	1	4	0,0001	-13,3	4,000	0,00010

$r(x)$ is the number of 1s in x

$$P(\mathbf{x}) = p_0^{N-r(\mathbf{x})} p_1^{r(\mathbf{x})}$$

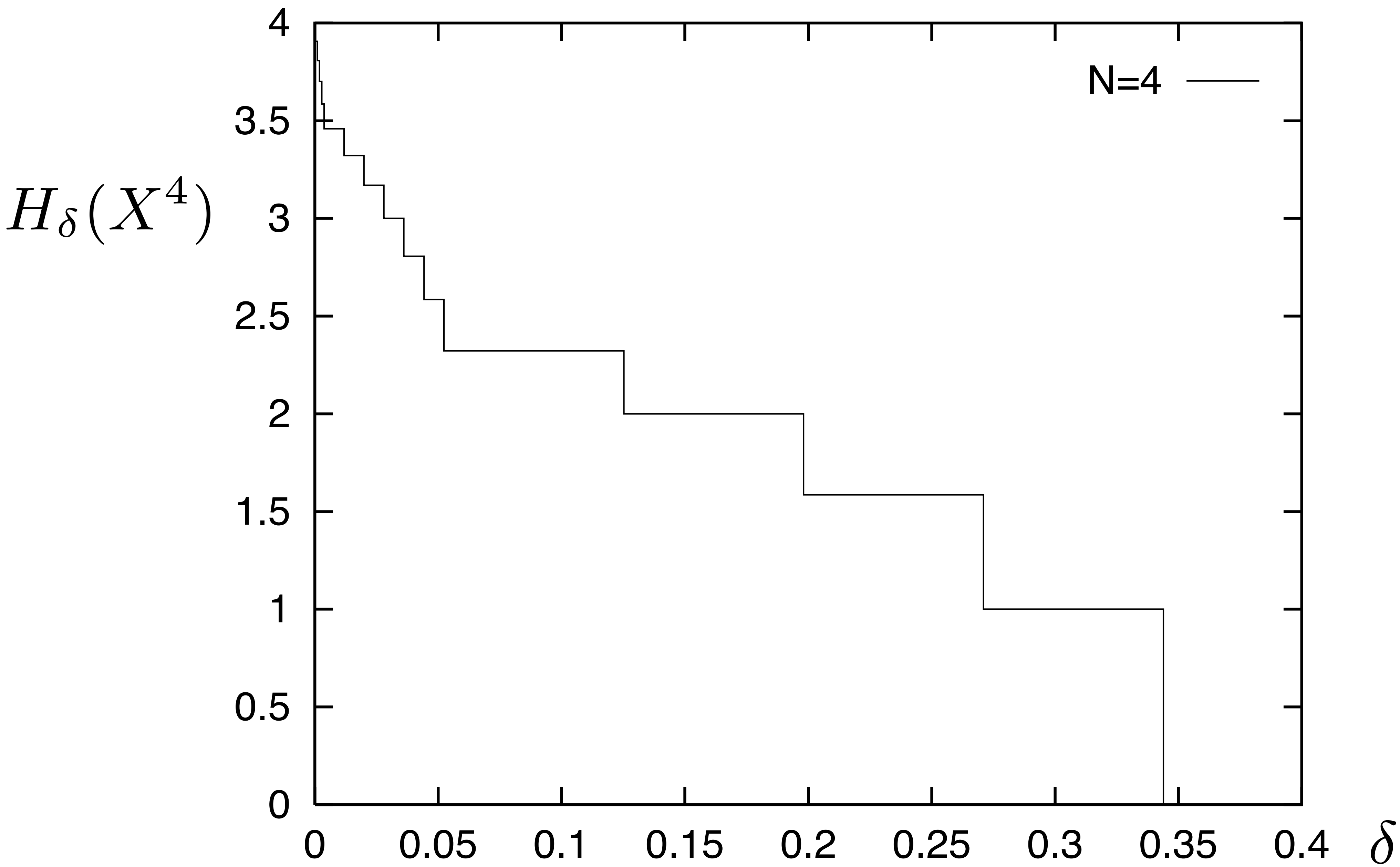
$p_0 = 0.9$ and $p_1 = 0.1$

$$H_\partial(X) = \log_2 |S_\partial|$$

$$P(x \in S_\partial) \geq 1 - \partial$$

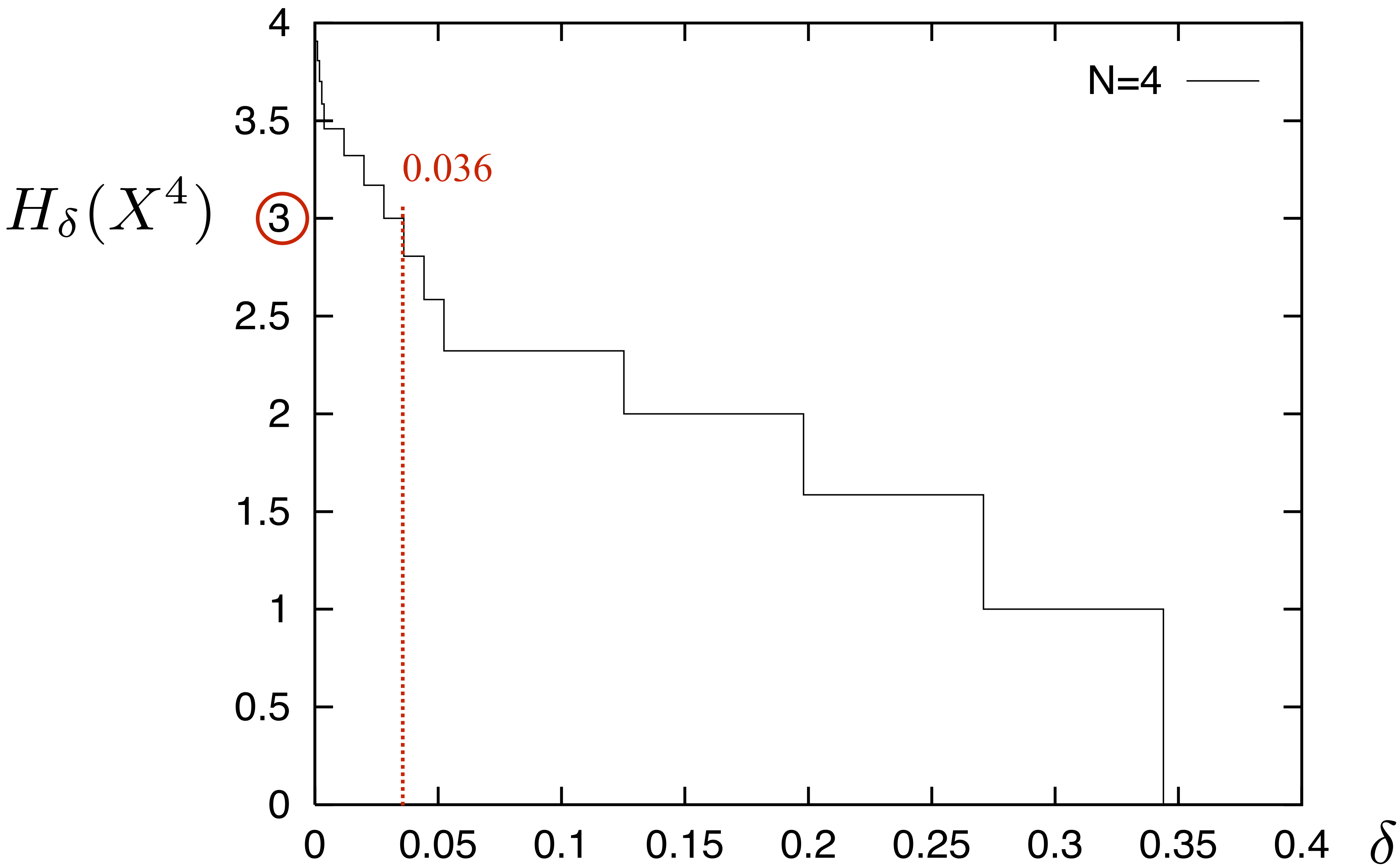
Extended ensembles - Example for $N = 4$

- To evaluate $H_\partial(X^N)$ we must find the smallest sufficient subset S_δ



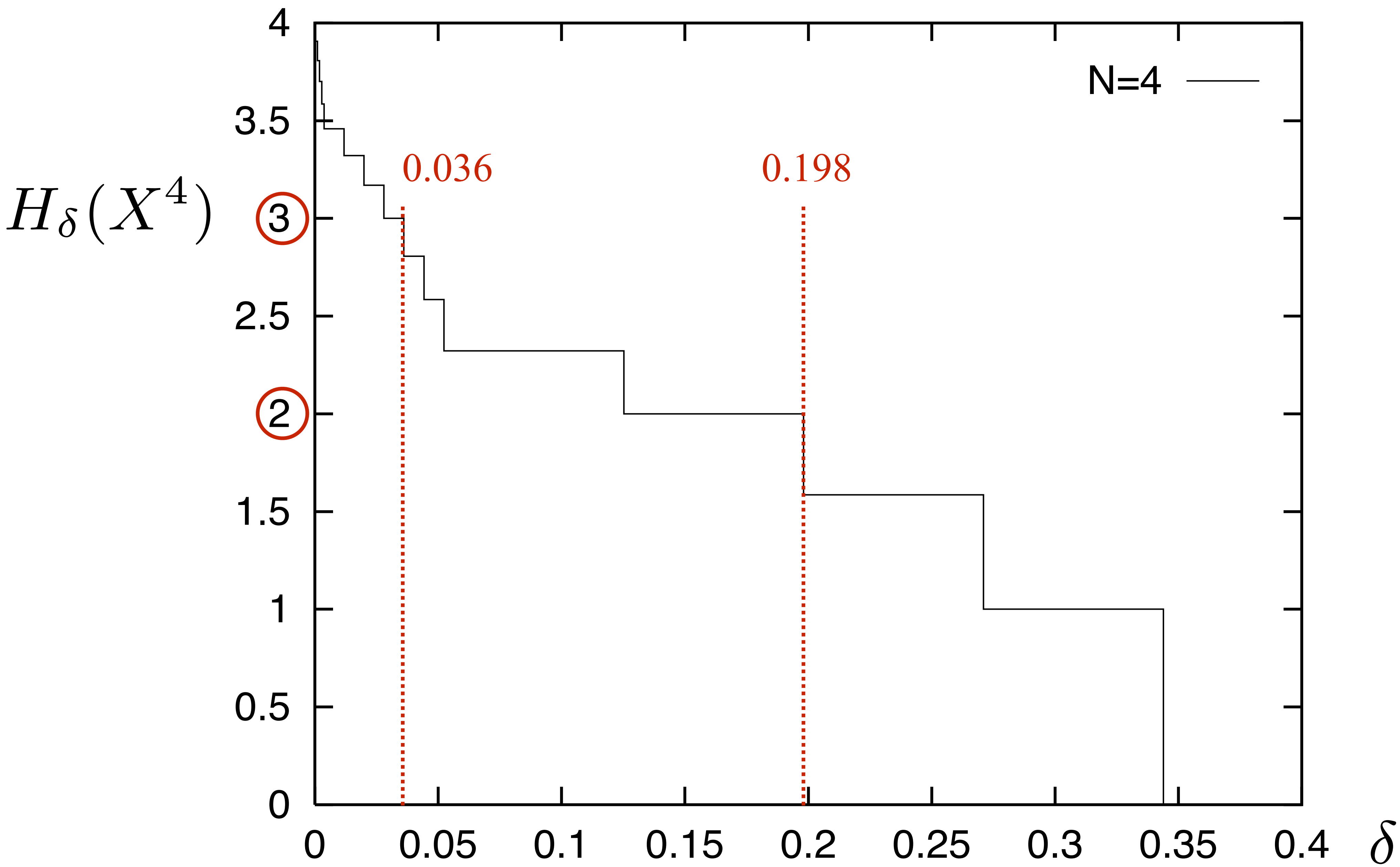
Extended ensembles - Example for $N = 4$

- To evaluate $H_\delta(X^N)$ we must find the smallest sufficient subset S_δ



Extended ensembles - Example for $N = 4$

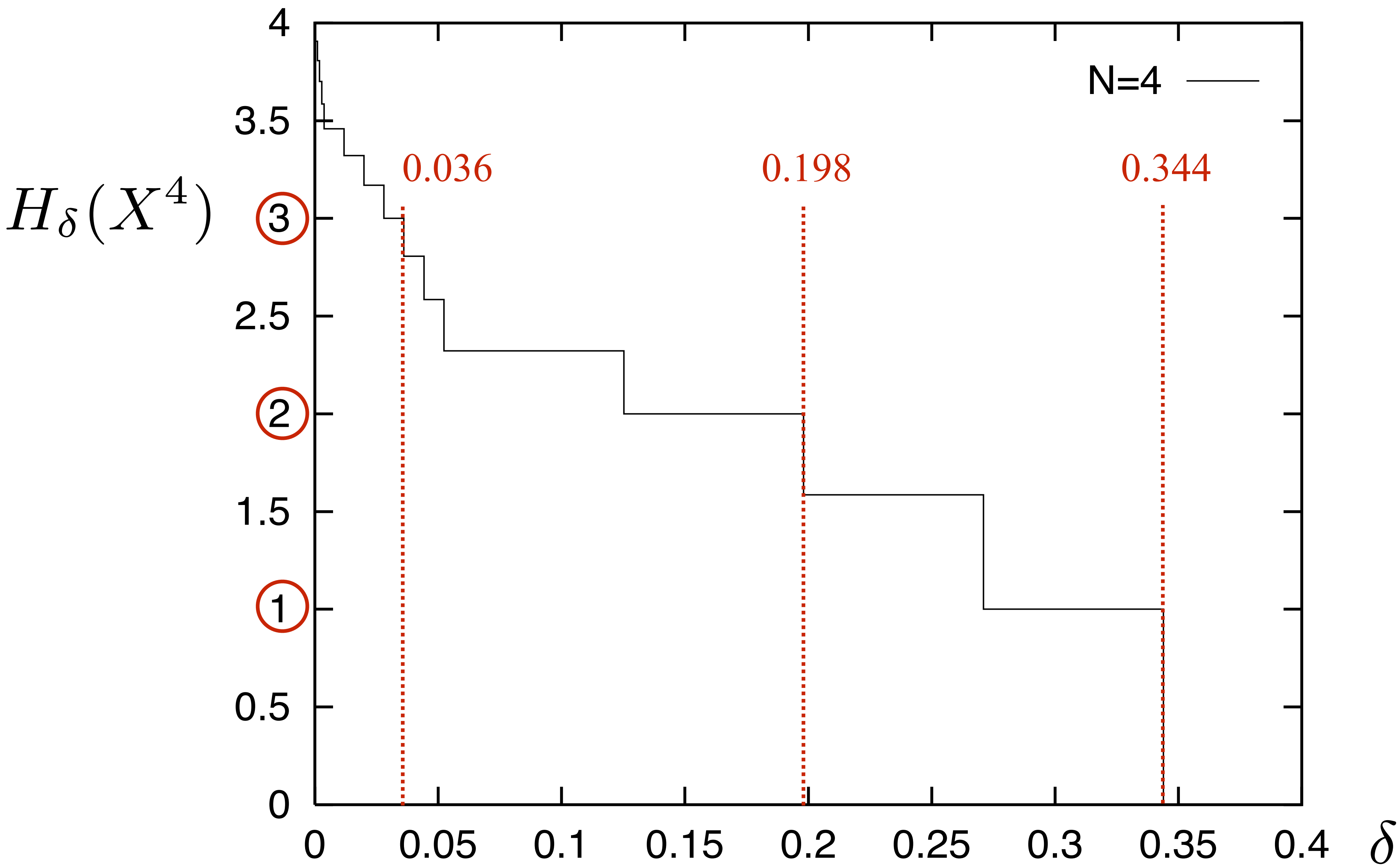
- To evaluate $H_\partial(X^N)$ we must find the smallest sufficient subset S_δ



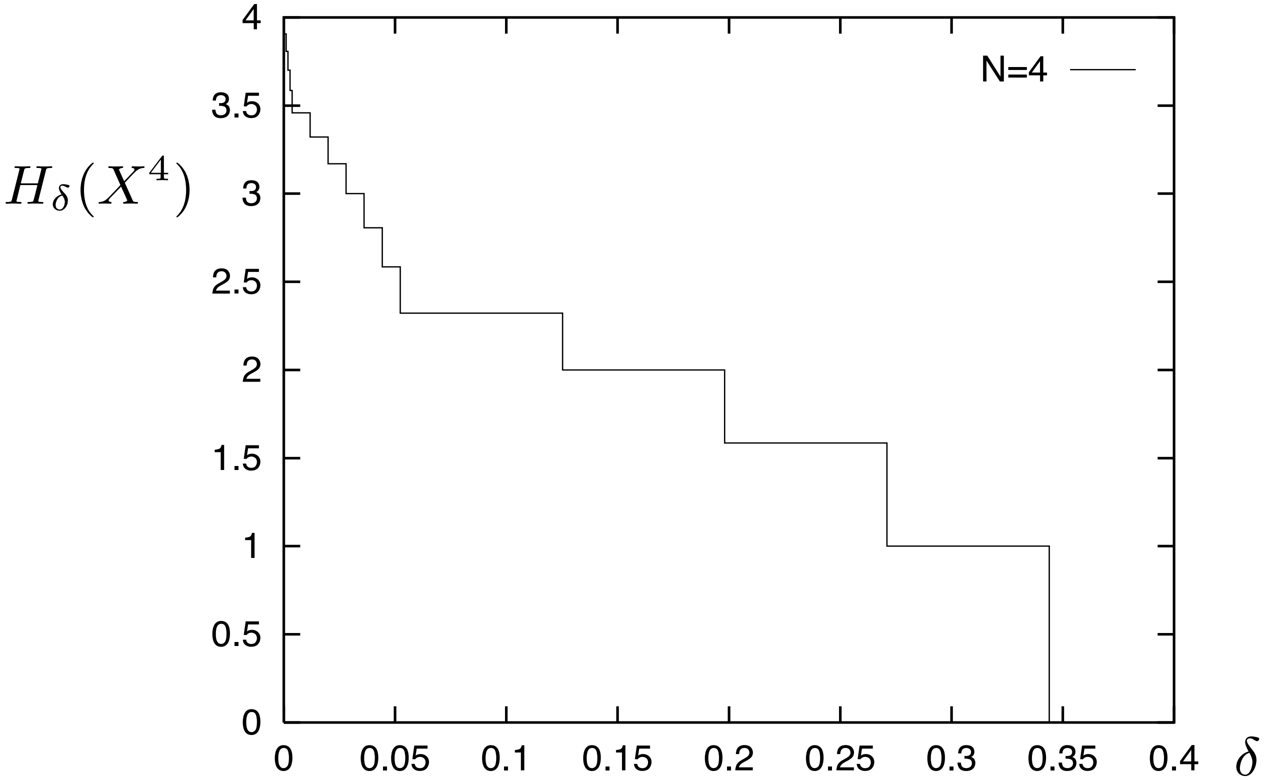
FACULDADE DE CIÊNCIAS E TECNOLOGIA UNIVERSIDADE NOVA DE LISBOA

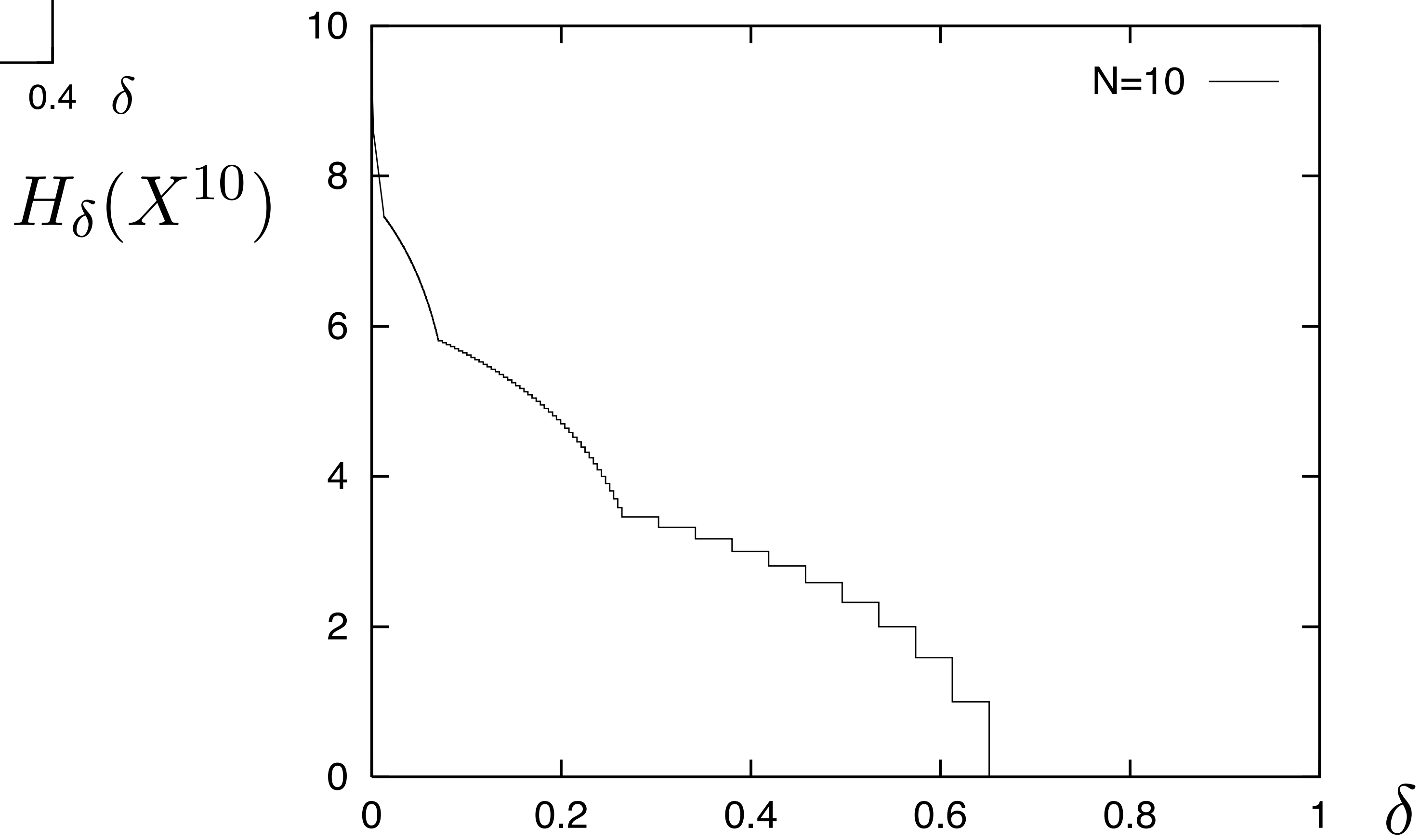
Extended ensembles - Example for $N = 4$

- To evaluate $H_\partial(X^N)$ we must find the smallest sufficient subset S_δ

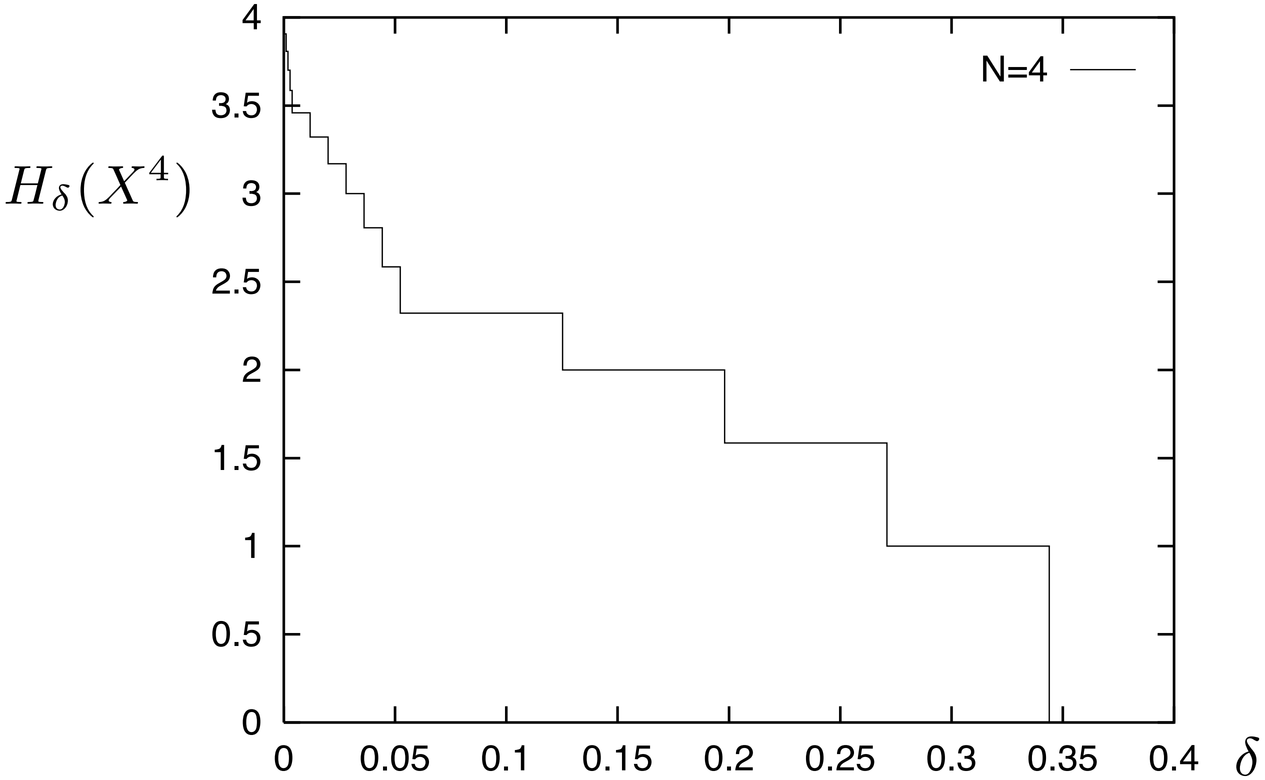


Extended ensembles - Example for N = 4 and N = 10

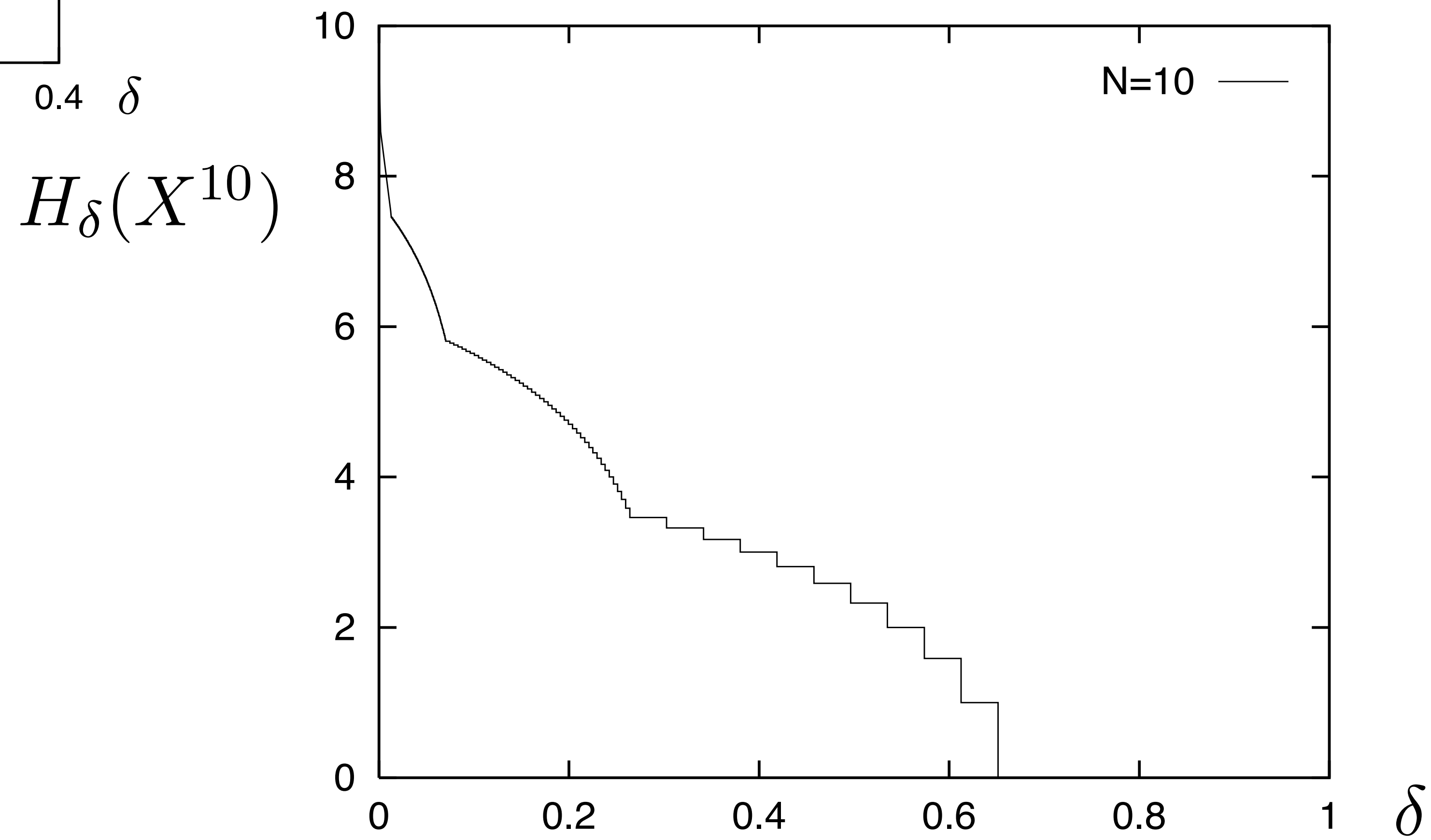




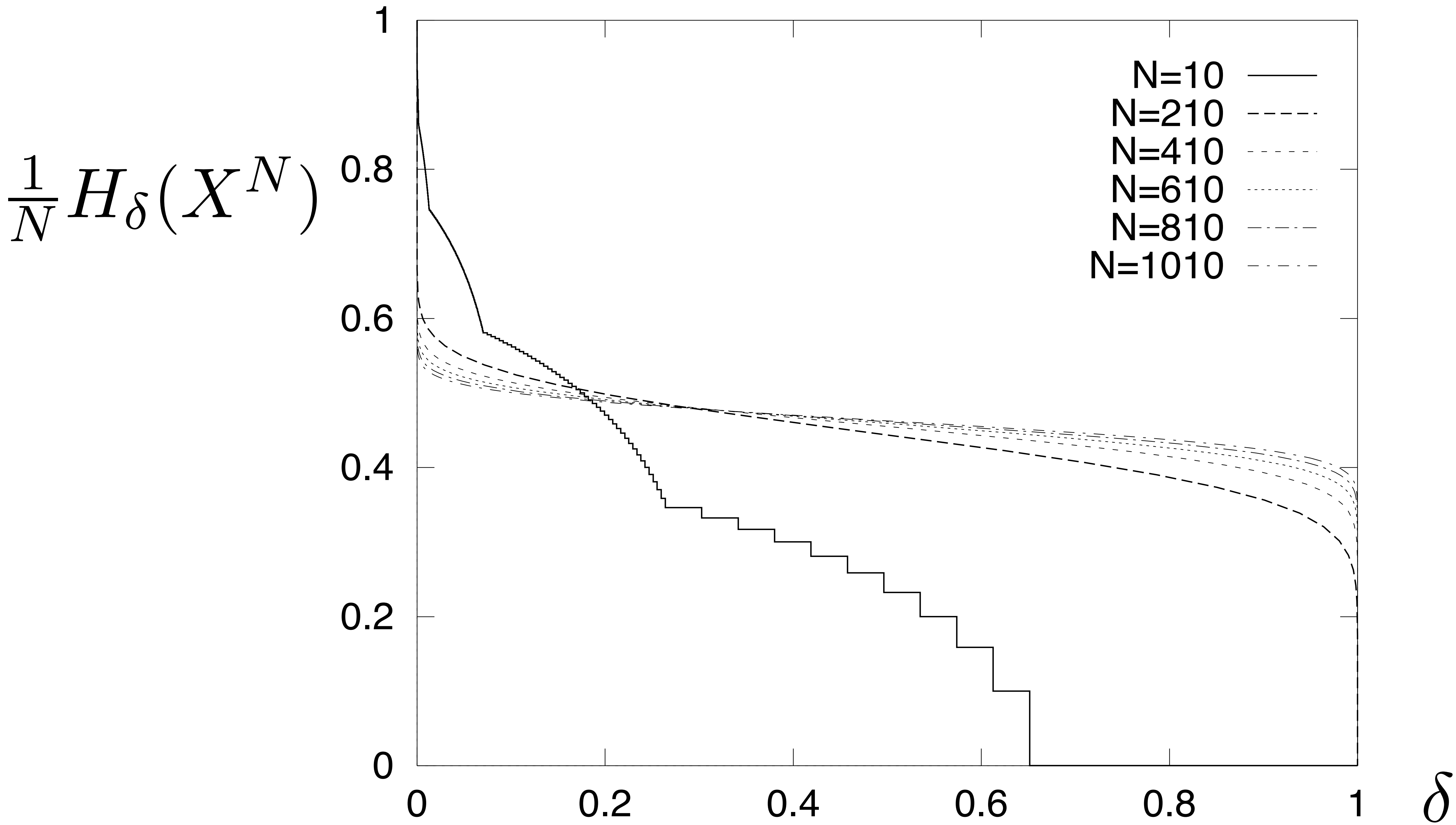
Extended ensembles - Example for N = 4 and N = 10



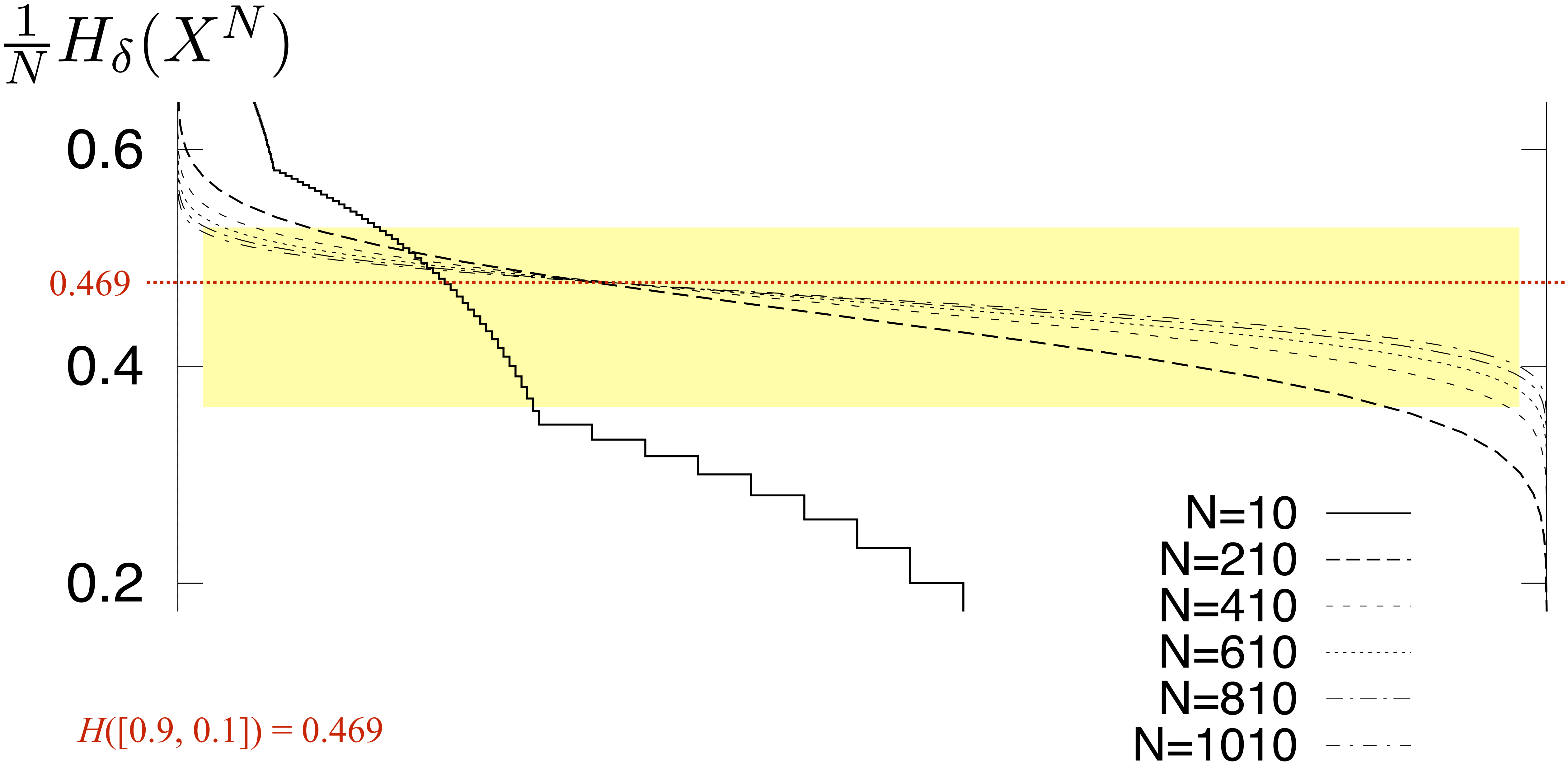
$H_\partial(X^N)$ depends strongly on values of ∂



Extended ensembles - N = 10, 210, 410, …, 1010



Extended ensembles - N = 10, 210, 410, …, 1010

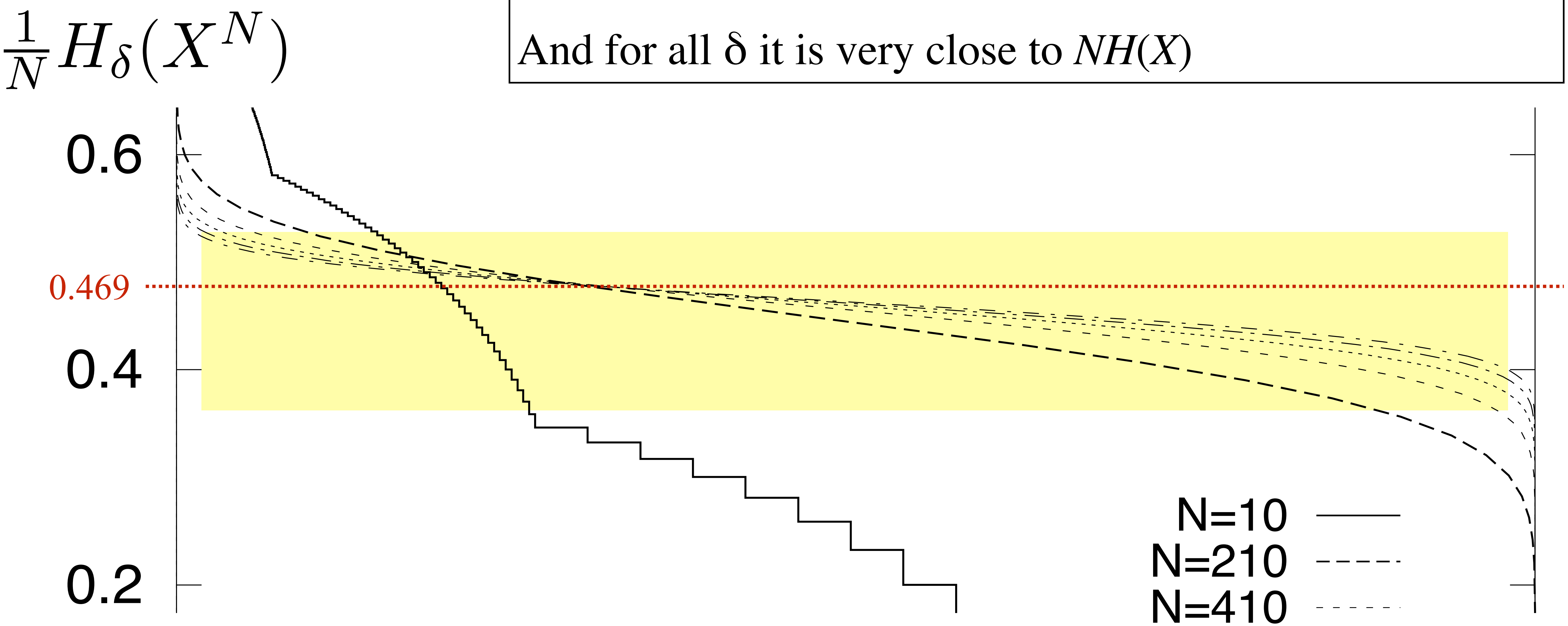


$H([0.9, 0.1]) = 0.469$

Extended ensembles - *N* = 10, 210, 410, …, 1010

As *N* increases $H_\partial(X^N)$ **becomes almost independent of δ**

And for all δ it is very close to *NH*(*X*)



H([0.9, 0.1]) = 0.469

Shannon's source coding theorem

Shannon's source coding theorem

- As long as we are **allowed a tiny probability of error δ**, **compression down to *N H* bits** is **possible**.

Shannon's source coding theorem

- As long as we are **allowed a tiny probability of error δ**, **compression down to *N H* bits** is **possible**.
- Even if we are **allowed a large probability of error**, we still **can compress only down to *N H*** bits.

Shannon's source coding theorem

- As long as we are **allowed a tiny probability of error δ**, **compression down to *N H* bits** is **possible**.
- Even if we are **allowed a large probability of error**, we still **can compress only down to *N H*** bits.

- **Shannon's source coding theorem**. Let X be an ensemble with entropy $H(X)$ = H bits. Given $\varepsilon > 0$ and $0 < \delta < 1$, there exists a positive integer N_0 such that for $N > N_0$,

$$\left| \frac{1}{N} H_{\partial}(X^N) - H \right| < \varepsilon$$

Typicality

Why does increasing N help?

- 15 samples from X^N for $N = 100$ and $p_1 = 0.1$, $p_0 = 0.9$.

$$H(X^N) = 46.9 \text{ bits}$$

Why does increasing N help?

- 15 samples from X^N for $N = 100$ and $p_1 = 0.1\,,\, p_0 = 0.9$.

$H(X^N) = 46.9$ bits

$\mathbf{x}$	$\log_2(P(\mathbf{x}))$
...1...................1.....1....1.1.......1........1...........1.....................1.......11...	−50.1
......................1.....1.....1.......1....1.........1.....................................1....	−37.3
........1....1..1...1....11..1.1.........11.........................1...1.1..1...1................1.	−65.9
1.1...1................1.......................11.1..1............................1.....1..1.11.....	−56.4
...11...........1...1.....1.1......1..........1....1...1.....1............1.........................	−53.2
..............1......1.........1.1.......1..........1............1...1......................1.......	−43.7
.....1........1.......1...1............1............1...........1......1..11........................	−46.8
.....1..1..1...............111...................1...............1.........1.1...1...1.............1	−56.4
.........1..........1.....1......1..........1....1..1...	−37.3
......1........................1..............1.....1..1.1.1..1...................................1.	−43.7
1.......................1..........1...1...................1....1....1........1..11..1.1...1........	−56.4
...........11.1.........1................1......1.....................1.............................	−37.3
.1..........1...1.1.............1.......11...........1.1...1..............1.............11..........	−56.4
......1...1..1.....1..11.1.1.1...1.....................1............1.............1..1..............	−59.5
............11.1......1....1..1............................1.......1..............1.......1.........	−46.8

Why does increasing N help?

- The most probable and the less probable sequences

$$H(X^N) = 46.9 \text{ bits}$$

$\mathbf{x}$	$\log_2(P(\mathbf{x}))$
...1...................1.....1....1.1.......1........1...........1.....................1.......11...	−50.1
......................1.....1.....1.......1....1.........1.....................................1....	−37.3
........1....1..1...1....11..1.1.........11.........................1...1.1..1...1................1.	−65.9
1.1...1................1.......................11.1..1............................1.....1..1.11.....	−56.4
...11...........1...1.....1.1......1..........1....1...1.....1............1.........................	−53.2
..............1......1.........1.1.......1..........1............1...1......................1.......	−43.7
.....1........1.......1...1............1............1...........1......1..11........................	−46.8
.....1..1..1...............111...................1...............1.........1.1...1...1.............1	−56.4
.........1..........1.....1......1..........1....1..1...	−37.3
......1........................1..............1.....1..1.1.1..1...................................1.	−43.7
1.......................1..........1...1...................1....1....1........1..11..1.1...1........	−56.4
...........11.1.........1................1......1.....................1.............................	−37.3
.1..........1...1.1.............1.......11...........1.1...1..............1.............11..........	−56.4
......1...1..1.....1..11.1.1.1...1.....................1............1..............1..1.............	−59.5
............11.1......1....1..1............................1.......1..............1........1........	−46.8
....	−15.2
1111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111111	−332.1

Why does increasing *N* help?

Why does increasing *N* help?

- The **probability** of a string $\boldsymbol{x}$ that contains r 1s and $N-r$ 0s is:

$$P(\boldsymbol{x}) = p_1^{\,r}(1-p_0)^{N-r}$$

Why does increasing *N* help?

- The **probability** of a string $\boldsymbol{x}$ that contains r 1s and $N-r$ 0s is:

$$P(\boldsymbol{x}) = p_1^{\,r}(1-p_0)^{N-r}$$

- The **number** of strings that contain r 1s is:

$$n(r) = \binom{N}{r}$$

Why does increasing *N* help?

- The **probability** of a string ***x*** that contains *r* 1s and *N*−*r* 0s is:

$$P(\boldsymbol{x}) = p_1^{\,r}(1 - p_0)^{N-r}$$

- The **number** of strings that contain *r* 1s is:

$$n(r) = \binom{N}{r}$$

- The number of 1s, *r*, has a **binomial distribution**:

$$P(r) = \binom{N}{r} p_1^{\,r}(1 - p_1)^{N-r}$$

Why does increasing *N* help?

- The **probability** of a string ***x*** that contains *r* 1s and *N*−*r* 0s is:

$$P(\boldsymbol{x}) = p_1^{\,r}(1 - p_0)^{N-r}$$

- The **number** of strings that contain *r* 1s is:

$$n(r) = \binom{N}{r}$$

- The number of 1s, *r*, has a **binomial distribution**:

$$P(r) = \binom{N}{r} p_1^{\,r}(1 - p_1)^{N-r}$$

$$\mu = Np_1$$

$$\sigma = \sqrt{Np_1(1 - p_1)}$$

Why does increasing *N* help?

- The **probability** of a string $\boldsymbol{x}$ that contains r 1s and $N-r$ 0s is:

$$P(\boldsymbol{x}) = p_1^{\,r}(1-p_0)^{N-r}$$

- The **number** of strings that contain r 1s is:

$$n(r) = \binom{N}{r}$$

- The number of 1s, r, has a **binomial distribution**:

$$P(r) = \binom{N}{r} p_1^{\,r}(1-p_1)^{N-r}$$

$$\mu = Np_1$$

$$\sigma = \sqrt{Np_1(1-p_1)}$$

For $N = 100$ and $p_1 = 0.1$

$$r \sim Np_1 \pm \sqrt{Np_1(1-p_1)} \simeq 10 \pm 3$$

Why does increasing N help?

- The **probability** of a string $\boldsymbol{x}$ that contains r 1s and $N-r$ 0s is:

$$P(\boldsymbol{x}) = p_1^{\,r}(1-p_1)^{N-r}$$

- The **number** of strings that contain r 1s is:

$$n(r) = \binom{N}{r}$$

- The number of 1s, r, has a **binomial distribution**:

$$P(r) = \binom{N}{r} p_1^{\,r}(1-p_1)^{N-r}$$

$$\mu = Np_1$$

$$\sigma = \sqrt{Np_1(1-p_1)}$$

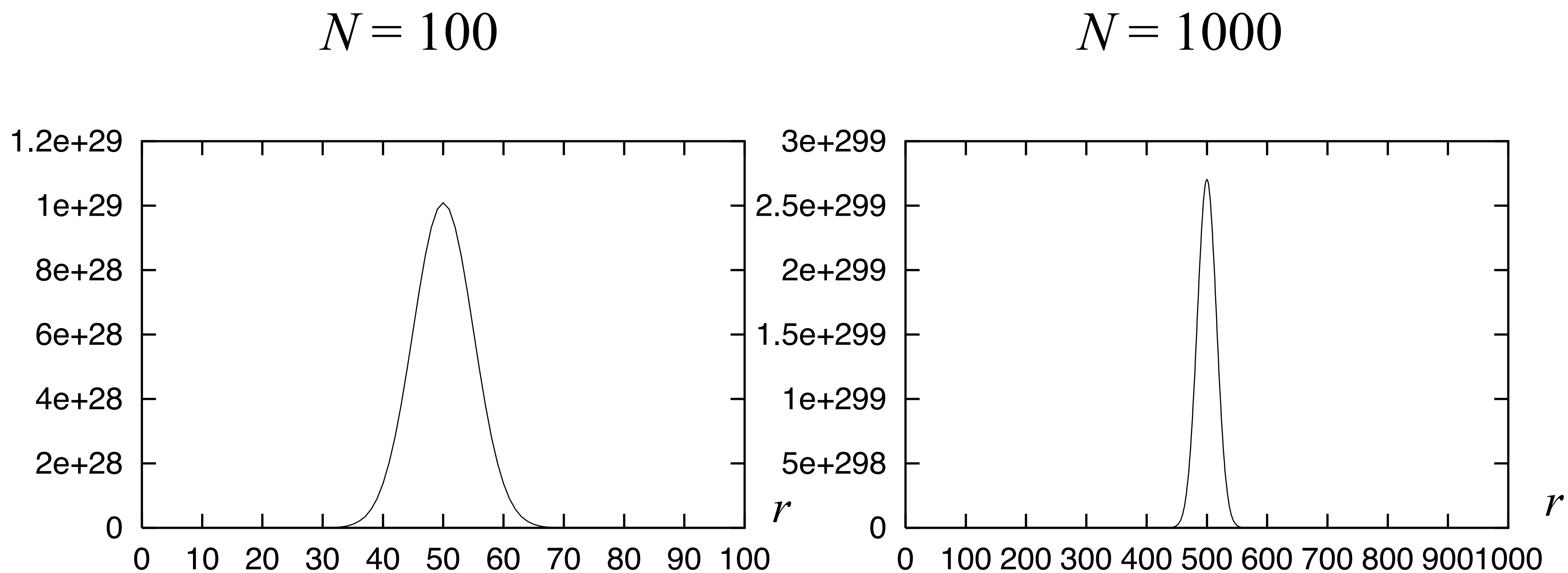
For $N = 1000$ and $p_1 = 0.1$

$$r \sim Np_1 \pm \sqrt{Np_1(1-p_1)} \simeq 100 \pm 10$$

Why does increasing N help?

For $p_1 = 0.1$

$$n(r) = \binom{N}{r}$$

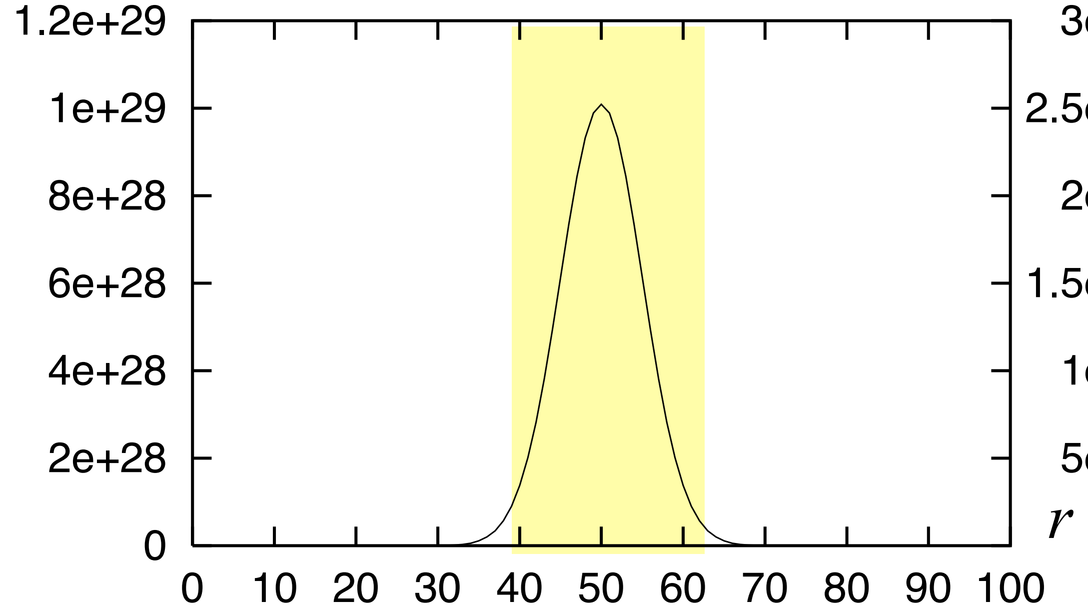


Why does increasing N help?

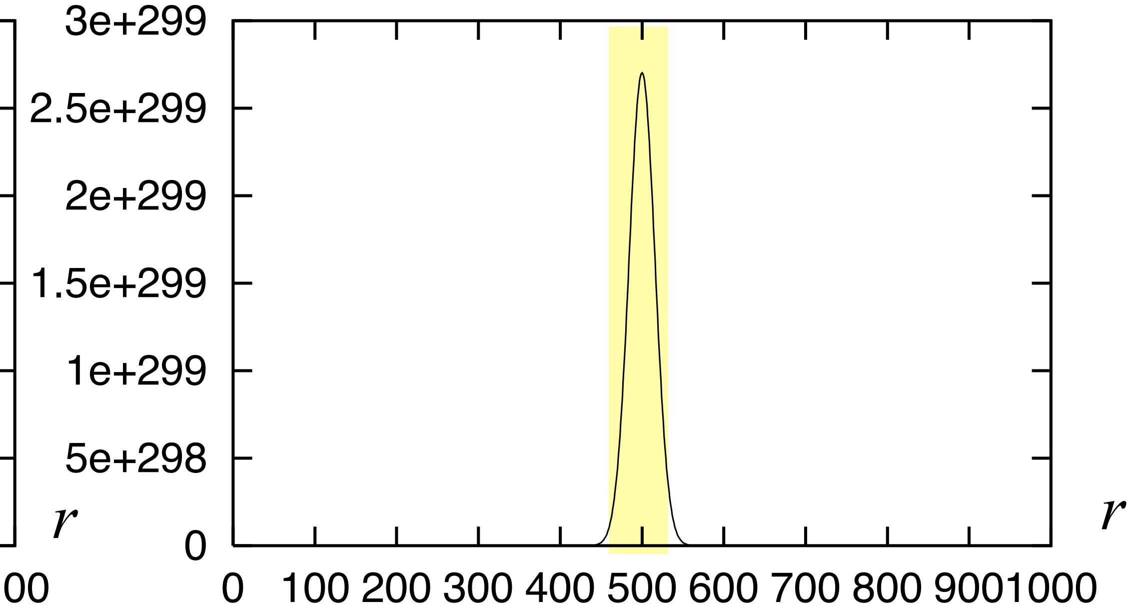
For $p_1 = 0.1$

$$n(r) = \binom{N}{r}$$

$N = 100$

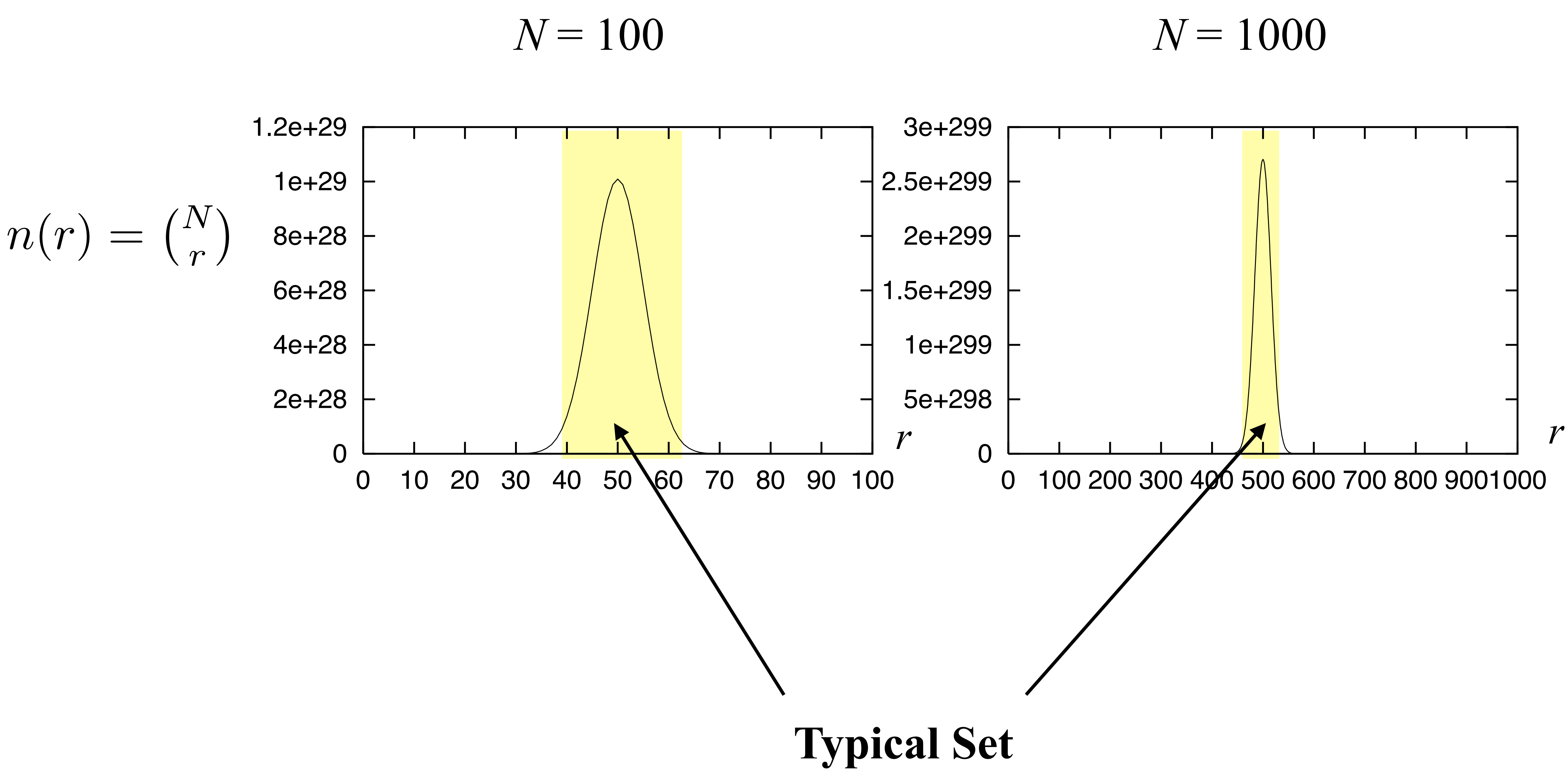


$N = 1000$



Why does increasing *N* help?

For $p_1 = 0.1$

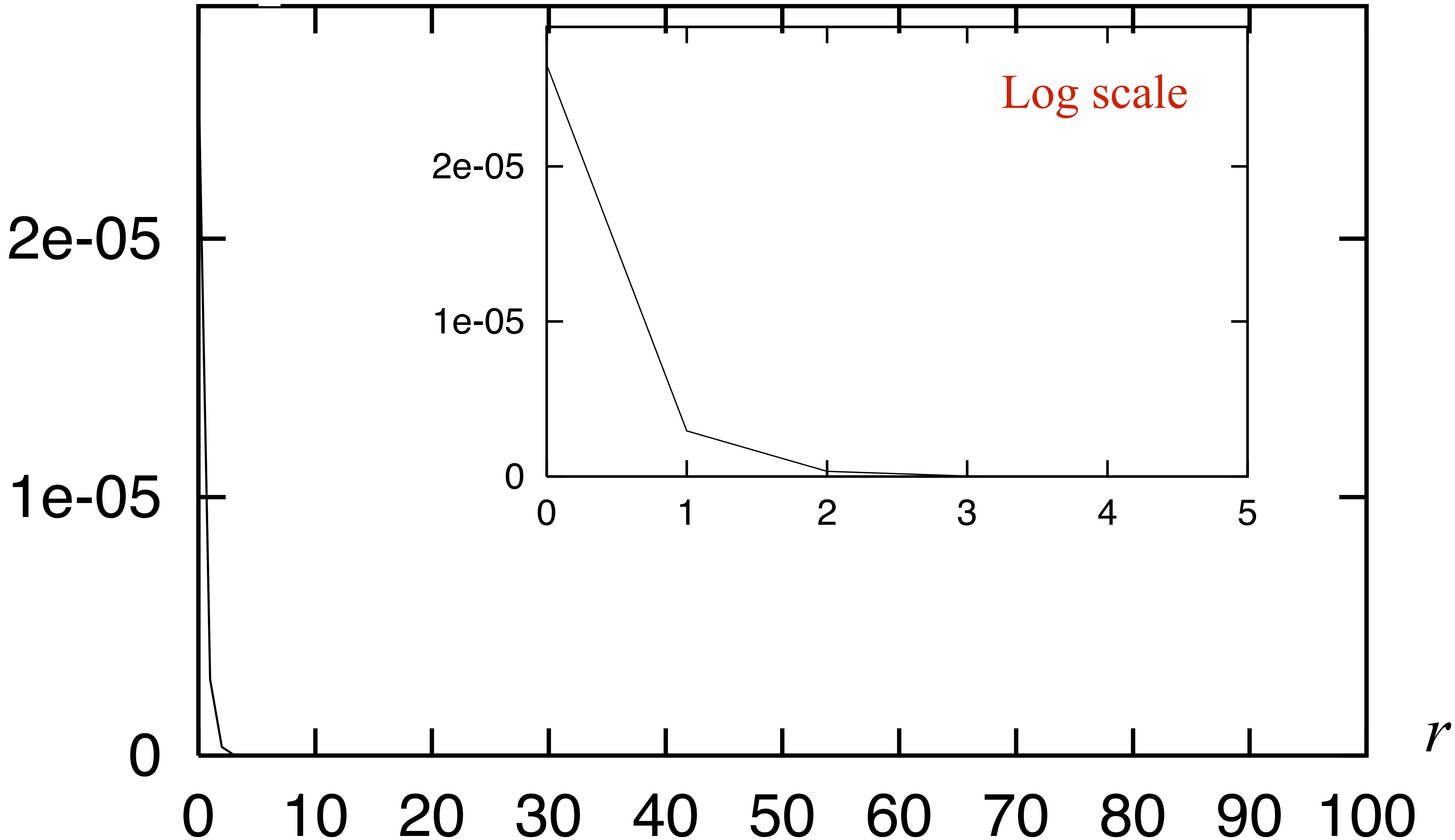


Why does increasing N help?

For $p_1 = 0.1$

$$P(\boldsymbol{x}) = p_1^{\,r}(1 - p_1)^{N-r}$$

$N = 100$

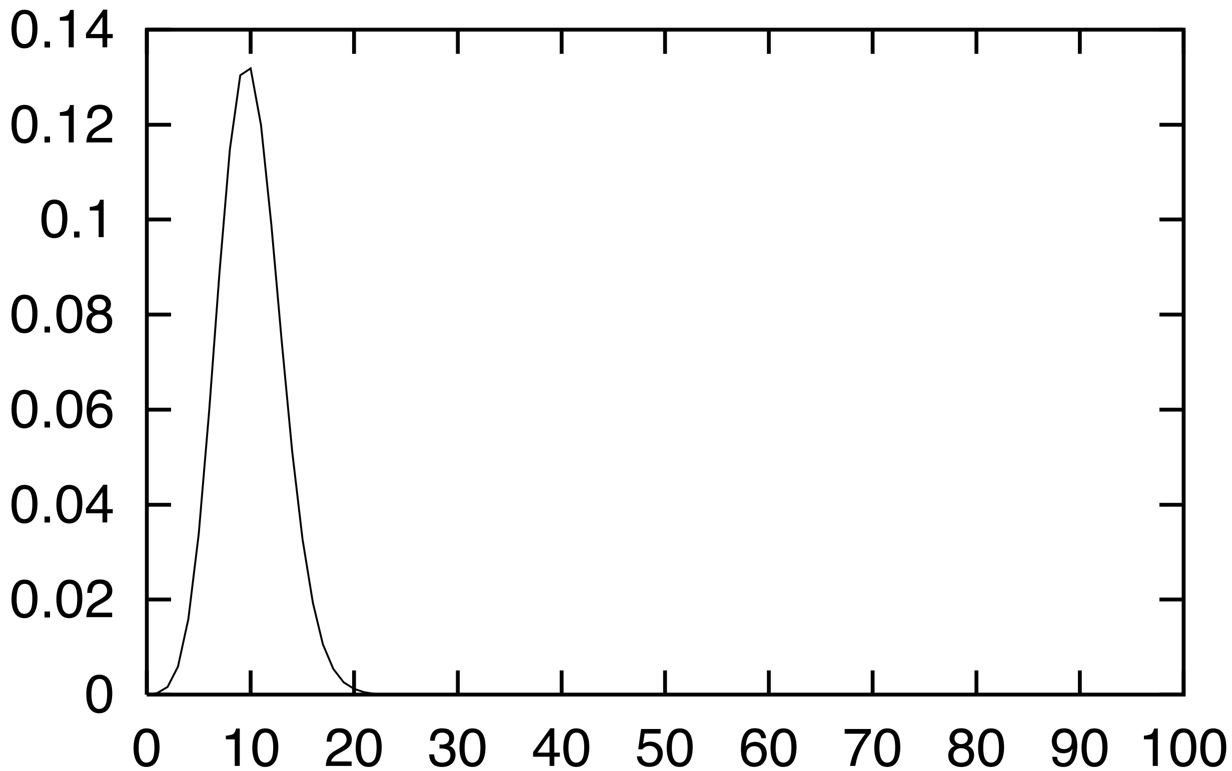


Why does increasing N help?

For $p_1 = 0.1$

$$n(r)P(\mathbf{x}) = \binom{N}{r} p_1^r (1 - p_1)^{N-r}$$

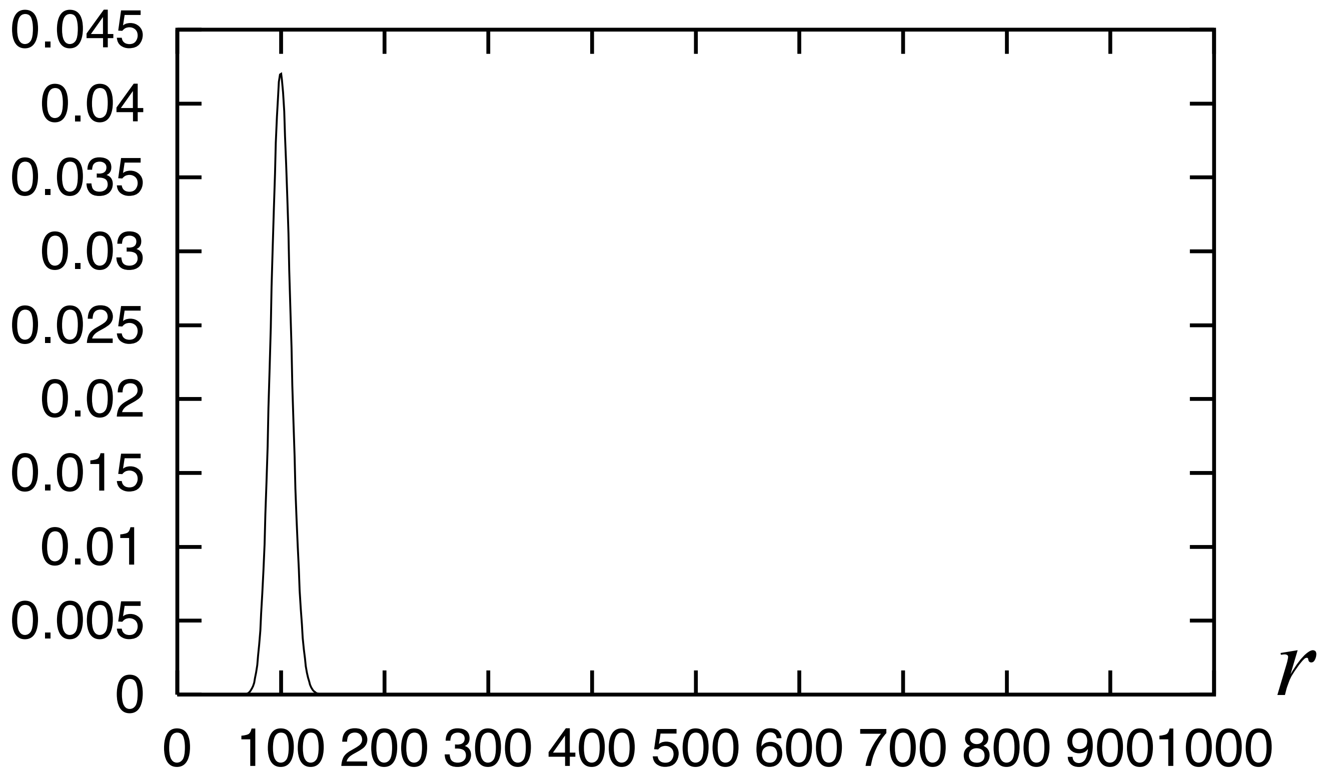
$N = 100$



For $N = 100$ and $p_1 = 0.1$

$$r \sim Np_1 \pm \sqrt{Np_1(1-p_1)} \simeq 10 \pm 3$$

$N = 1000$



For $N = 1000$ and $p_1 = 0.1$

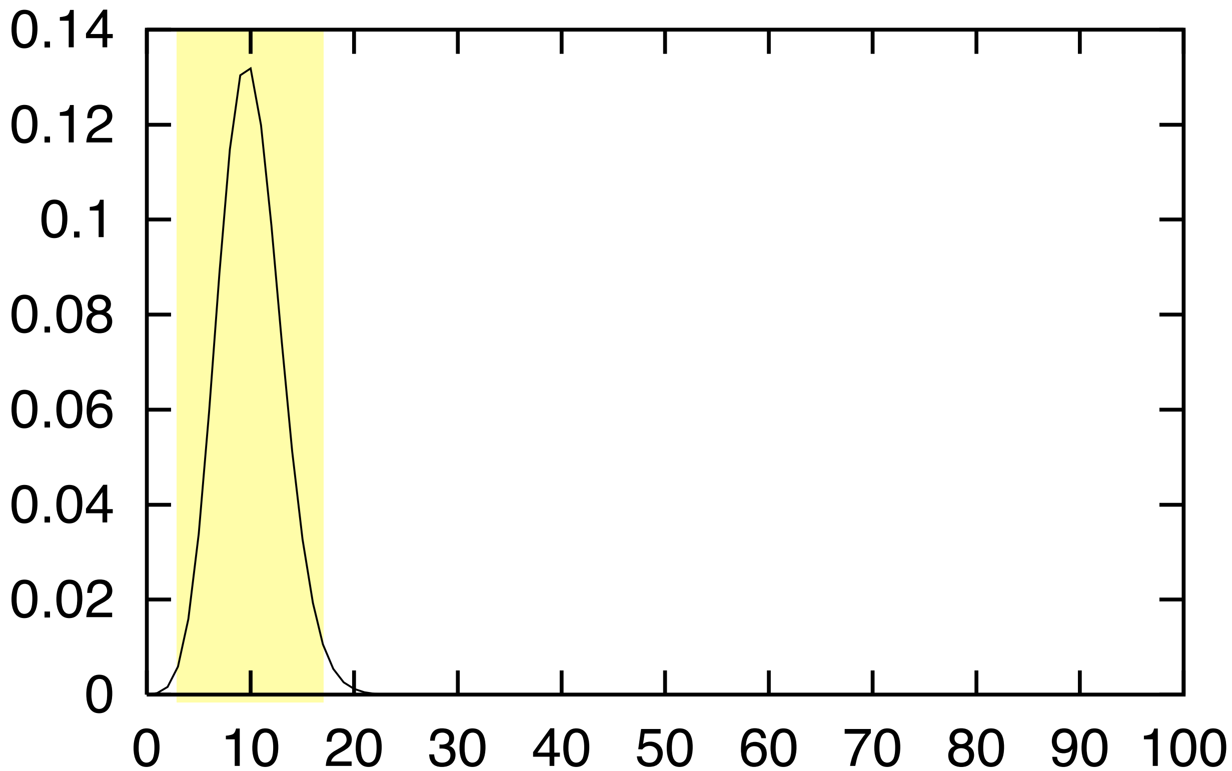
$$r \sim Np_1 \pm \sqrt{Np_1(1-p_1)} \simeq 100 \pm 10$$

Why does increasing N help?

For $p_1 = 0.1$

$$n(r)P(\mathbf{x}) = \binom{N}{r} p_1^r (1-p_1)^{N-r}$$

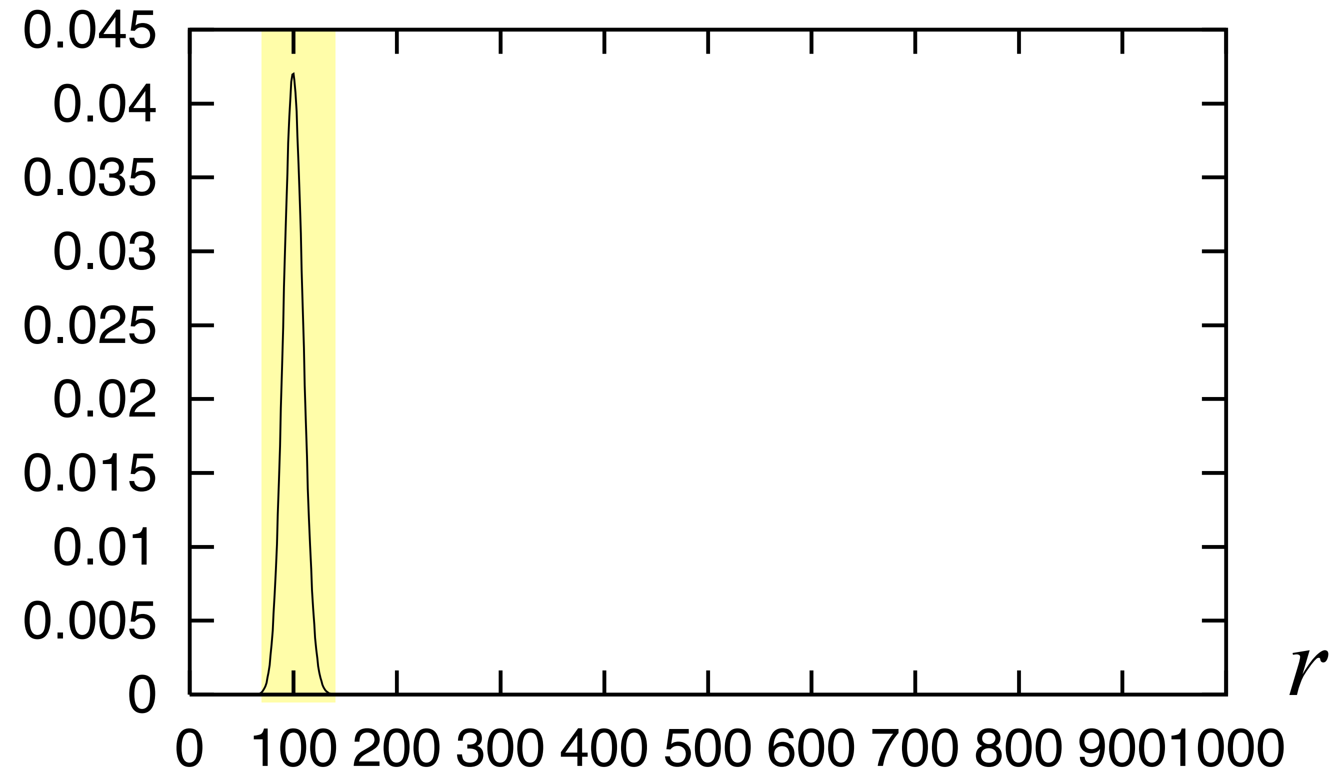
$N = 100$



For $N = 100$ and $p_1 = 0.1$

$$r \sim Np_1 \pm \sqrt{Np_1(1-p_1)} \simeq 10 \pm 3$$

$N = 1000$



For $N = 1000$ and $p_1 = 0.1$

$$r \sim Np_1 \pm \sqrt{Np_1(1-p_1)} \simeq 100 \pm 10$$

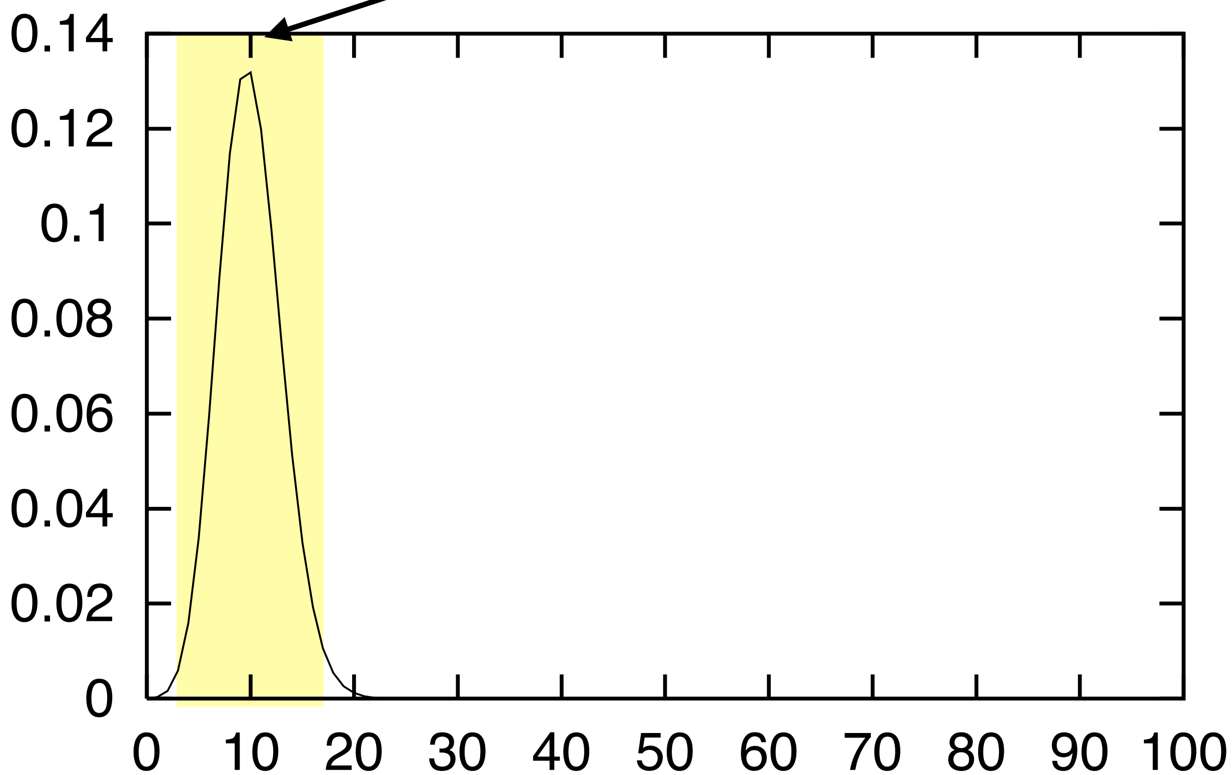
Why does increasing N help?

For $p_1 = 0.1$

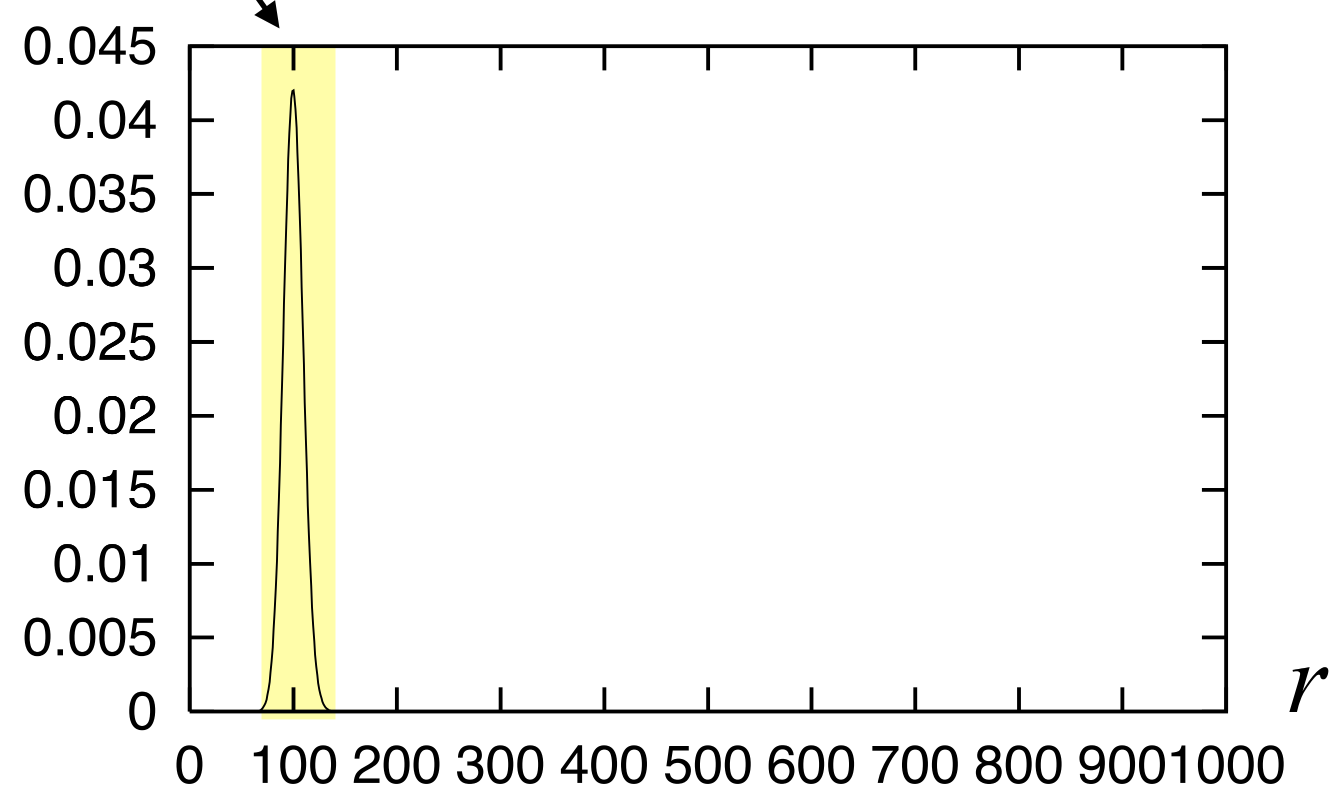
$$n(r)P(\mathbf{x}) = \binom{N}{r} p_1^r (1 - p_1)^{N-r}$$

Typical Set

$N = 100$



$N = 1000$



For $N = 100$ and $p_1 = 0.1$

$$r \sim Np_1 \pm \sqrt{Np_1(1 - p_1)} \simeq 10 \pm 3$$

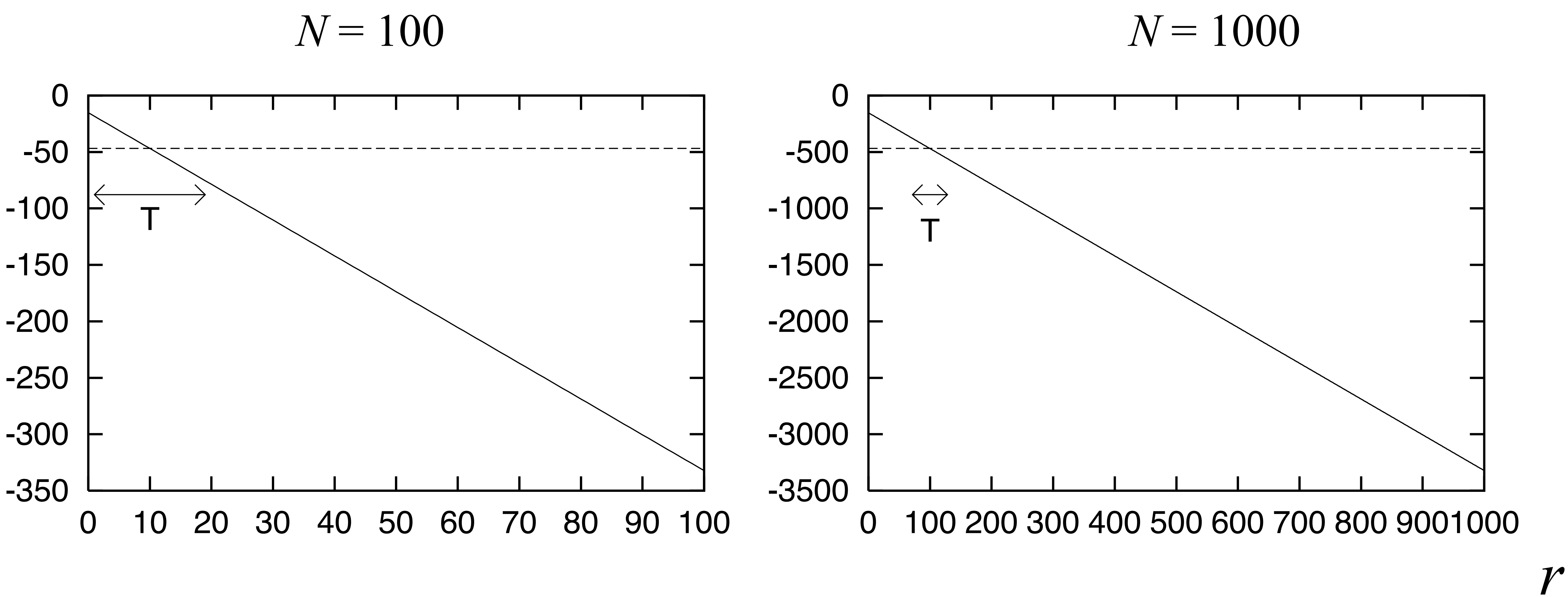
For $N = 1000$ and $p_1 = 0.1$

$$r \sim Np_1 \pm \sqrt{Np_1(1 - p_1)} \simeq 100 \pm 10$$

Why does increasing N help?

For $p_1 = 0.1$

$\log_2 P(\mathbf{x})$

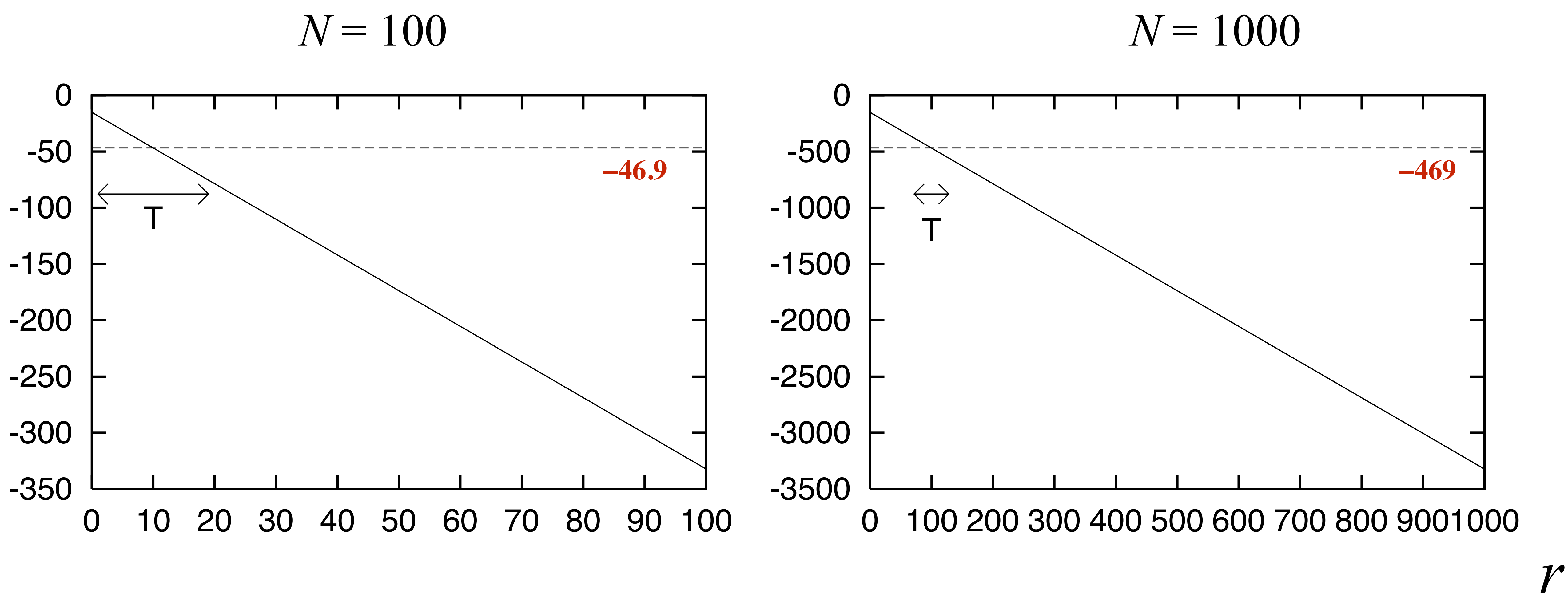


Why does increasing N help?

For $p_1 = 0.1$

$\log_2 P(\mathbf{x})$

Mean of $\log_2 P(\boldsymbol{x}) = -NH_2(p_1)$

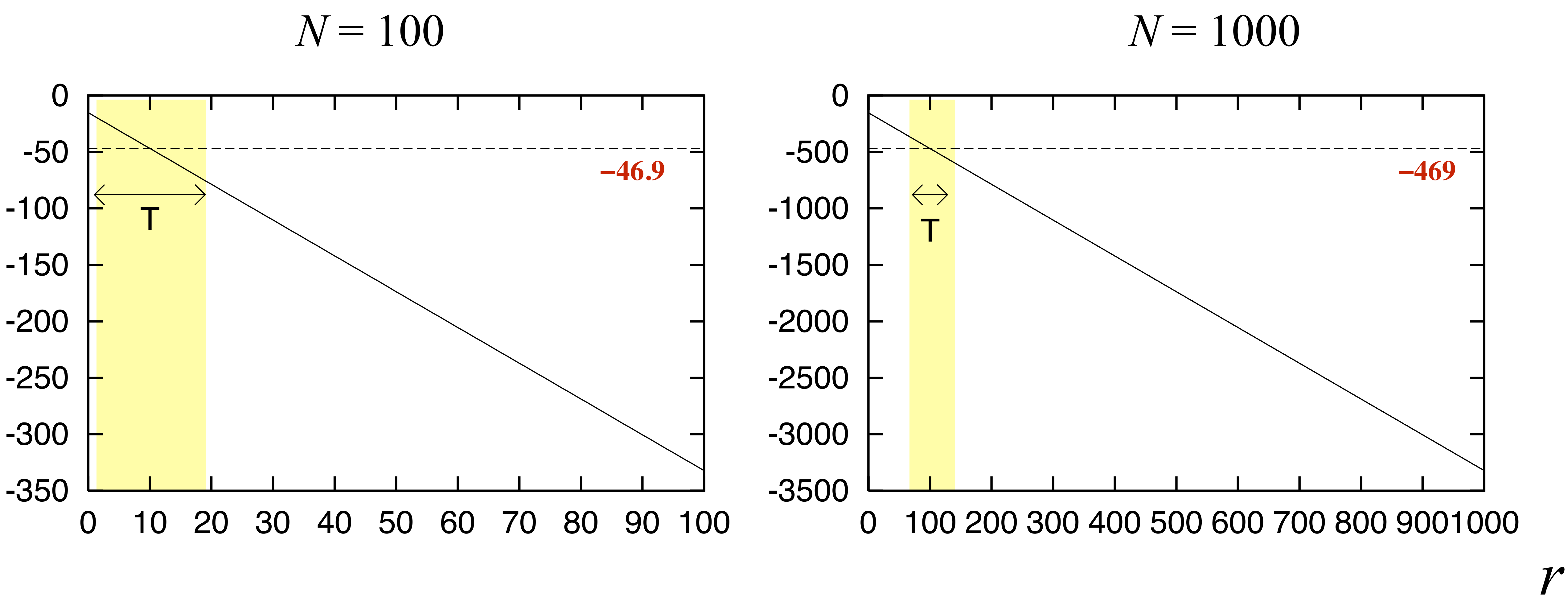


Why does increasing N help?

For $p_1 = 0.1$

$\log_2 P(\mathbf{x})$

Mean of $\log_2 P(\mathbf{x}) = -NH_2(p_1)$



The **typical set** includes only the strings that have $\log_2 P(\mathbf{x})$ close to this value $-NH_2(p_1)$

Definition of the typical set

- For an arbitrary ensemble X with alphabet $A_X = \{x_1, \ldots, x_i, \ldots, x_I\}$, a long string of N symbols will **usually** contain about $\boldsymbol{p_1 N}$ occurrences of the first symbol, $\boldsymbol{p_2 N}$ occurrences of the second, … $\boldsymbol{p_I N}$ occurrences of the last symbol.

$$P(\mathbf{x}) = P(x_1)P(x_2)P(x_3)\ldots P(x_N)$$

Definition of the typical set

- For an arbitrary ensemble X with alphabet $A_X = \{x_1, \ldots, x_i, \ldots, x_I\}$, a long string of N symbols will **usually** contain about $\boldsymbol{p_1 N}$ occurrences of the first symbol, $\boldsymbol{p_2 N}$ occurrences of the second, … $\boldsymbol{p_I N}$ occurrences of the last symbol.

$$P(\boldsymbol{x}) = P(x_1)P(x_2)P(x_3)\ldots P(x_N) \simeq p_1^{\,p_1 N} p_2^{\,p_2 N} \ldots p_I^{\,p_I N}$$

Definition of the typical set

- For an arbitrary ensemble X with alphabet $A_X = \{x_1, \ldots, x_i, \ldots, x_I\}$, a long string of N symbols will **usually** contain about $\boldsymbol{p_1 N}$ occurrences of the first symbol, $\boldsymbol{p_2 N}$ occurrences of the second, … $\boldsymbol{p_I N}$ occurrences of the last symbol.

$$P(\boldsymbol{x}) = P(x_1)P(x_2)P(x_3)\ldots P(x_N) \simeq p_1^{\,p_1 N} p_2^{\,p_2 N} \ldots p_I^{\,p_I N}$$

- So the information content **of a typical string** is

$$\log_2 \frac{1}{P(\boldsymbol{x})} \simeq N \sum_i p_i \log_2 \frac{1}{p_i}$$

Definition of the typical set

- For an arbitrary ensemble X with alphabet $A_X = \{x_1, \ldots, x_i, \ldots, x_I\}$, a long string of N symbols will **usually** contain about $\boldsymbol{p_1 N}$ occurrences of the first symbol, $\boldsymbol{p_2 N}$ occurrences of the second, … $\boldsymbol{p_I N}$ occurrences of the last symbol.

$$P(\boldsymbol{x}) = P(x_1)P(x_2)P(x_3)\ldots P(x_N) \simeq p_1^{\,p_1 N} p_2^{\,p_2 N} \ldots p_I^{\,p_I N}$$

- So the information content **of a typical string** is

$$\log_2 \frac{1}{P(\boldsymbol{x})} \simeq N \sum_i p_i \log_2 \tfrac{1}{p_i} = NH(X) = NH$$

Definition of the typical set

- For an arbitrary ensemble X with alphabet $A_X = \{x_1, \ldots, x_i, \ldots, x_I\}$, a long string of N symbols will **usually** contain about $\boldsymbol{p_1 N}$ occurrences of the first symbol, $\boldsymbol{p_2 N}$ occurrences of the second, … $\boldsymbol{p_I N}$ occurrences of the last symbol.

$$P(\boldsymbol{x}) = P(x_1)P(x_2)P(x_3)\ldots P(x_N) \simeq p_1^{\,p_1 N} p_2^{\,p_2 N} \ldots p_I^{\,p_I N}$$

- So the information content **of a typical string** is

$$\log_2 \frac{1}{P(\boldsymbol{x})} \simeq N \sum_i p_i \log_2 \tfrac{1}{p_i} = NH(X) = NH$$

- So, the random variable $\log_2 1/P(\mathbf{x})$, which is the information content of $\mathbf{x}$, is very likely to be close in value to NH.

Definition of the typical set

- So, the random variable $\log_2 1/P(\mathbf{x})$, which is the information content of $\mathbf{x}$, is very likely to be close in value to NH.

Definition of the typical set

- So, the random variable $\log_2 1/P(\mathbf{x})$, which is the information content of $\mathbf{x}$, is very likely to be close in value to NH.
- Lets define the **typical elements of** A_X^N to be those elements that have probability close to 2^{-NH}

Definition of the typical set

- So, the random variable $\log_2 1/P(\mathbf{x})$, which is the information content of $\mathbf{x}$, is very likely to be close in value to NH.
- Lets define the **typical elements of** A_X^N to be those elements that have probability close to 2^{-NH}
- Consider a parameter ß that defines how close the probability has to be to 2^{-NH} for an element be *typical*.

Definition of the typical set

- So, the random variable $\log_2 1/P(\mathbf{x})$, which is the information content of $\mathbf{x}$, is very likely to be close in value to NH.
- Lets define the **typical elements of** A_X^N to be those elements that have probability close to 2^{-NH}
- Consider a parameter ß that defines how close the probability has to be to 2^{-NH} for an element be *typical*.
- The typical set, $T_{N\beta}$:

$$T_{N\beta} = \left\{ \boldsymbol{x} \in A_X^N : \left| \frac{1}{N} \log_2 \frac{1}{P(\boldsymbol{x})} - H \right| < \beta \right\}$$

Definition of the typical set

- The typical set, $T_{N\beta}$:

- Notes
  - Unlike the **smallest sufficient subset**, the **typical set does not include the most probable elements** of A_X^N
  - But these most probable elements contribute negligible probability.
  - Whatever value of ß we choose, the **typical set contains almost all the probability as** ***N*** **increases**

FACULDADE DE CIÊNCIAS E TECNOLOGIA UNIVERSIDADE NOVA DE LISBOA

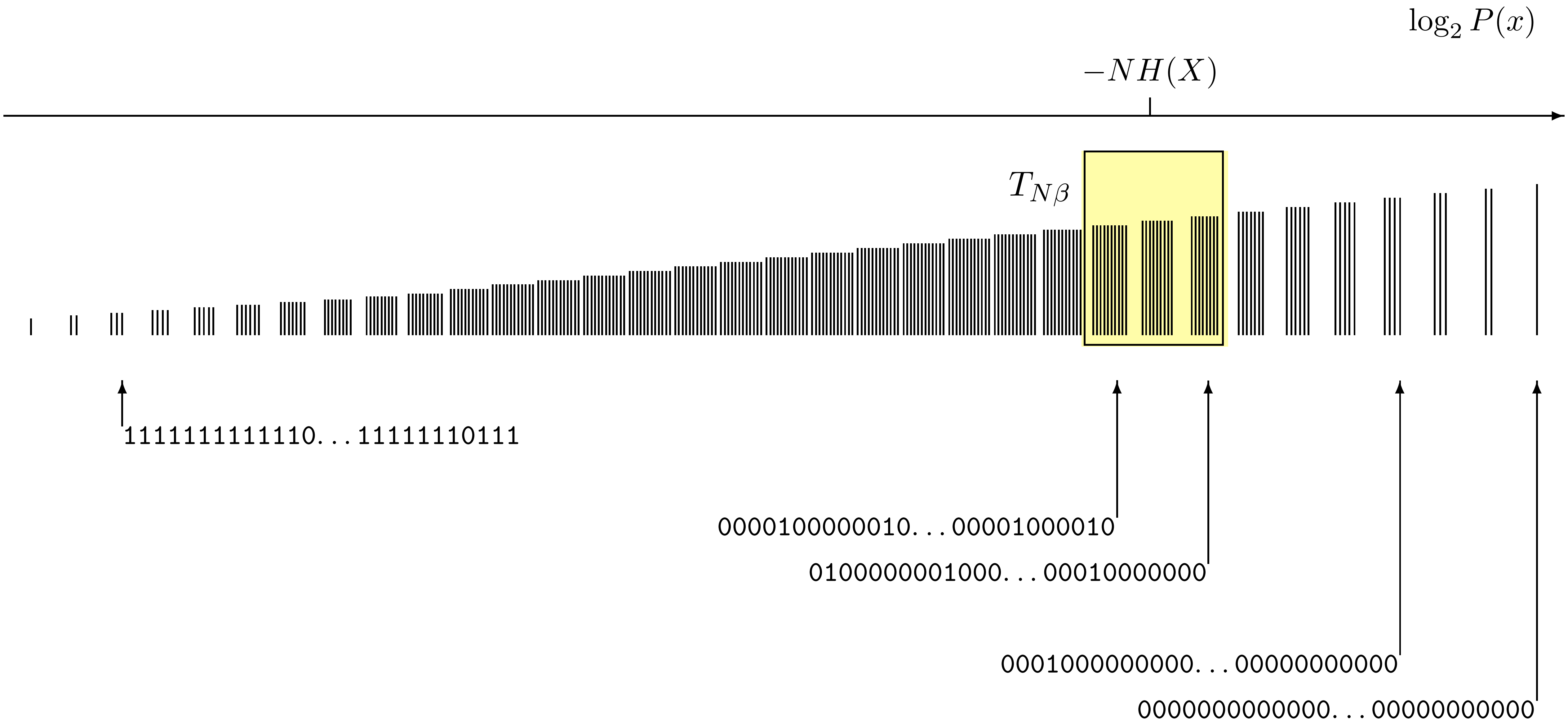
Definition of the typical set

- The typical set, $T_{N\beta}$:

$$T_{N\beta} = \left\{ \boldsymbol{x} \in A_X^N : \left| \frac{1}{N} \log_2 \frac{1}{P(\boldsymbol{x})} - H \right| < \beta \right\}$$

- Notes
  - Unlike the **smallest sufficient subset**, the **typical set does not include the most probable elements** of A_X^N
  - But these most probable elements contribute negligible probability.
  - Whatever value of ß we choose, the **typical set contains almost all the probability as** N **increases**

all strings in the ensemble X^N



All strings in the ensemble X^N ranked by their probability

‘Asymptotic equipartition’ principle

- For an ensemble of N independent identically distributed (i.i.d.) random variables $X^N \equiv (X_1, X_2, \ldots, X_N)$, with N sufficiently large, the outcome $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ is almost certain to belong to a subset of A_X^N having only $2^{NH(X)}$ members, each having probability ‘close to’ $2^{-NH(X)}$.
- Notice that if $H(X) < H_0(X)$ then $2^{NH(X)}$ is a tiny fraction of the number of possible outcomes

$$\left|A_X^N\right| = \left|A_X\right|^N = 2^{NH_0(X)}$$

‘Asymptotic equipartition’ principle

- For an ensemble of N independent identically distributed (i.i.d.) random variables $X^N \equiv (X_1, X_2, \ldots, X_N)$, with N sufficiently large, the outcome $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ is almost certain to belong to a subset of A_X^N having only $2^{NH(X)}$ members, each having probability ‘close to’ $2^{-NH(X)}$.
- Notice that if $H(X) < H_0(X)$ then $2^{NH(X)}$ is a tiny fraction of the number of possible outcomes

$$\left|A_X^N\right| = \left|A_X\right|^N = 2^{NH_0(X)}$$

$$H_0(X) = \log_2 \left|A_X\right|$$

Shannon's source coding theorem (verbal statement)

- N independent identically distributed random variables each with entropy $H(X)$ can be compressed into more than $NH(X)$ bits with negligible risk of information loss, as $N \to \infty$

- Conversely if they are compressed into fewer than $NH(X)$ bits it is virtually certain that information will be lost.

Comments on source coding theorem

Two parts of Shannon's source coding theorem

- The source coding theorem has **two parts** !

$$\left|\frac{1}{N}H_{\partial}(X^N)-H\right|<\varepsilon$$

- $\frac{1}{N}H_{\partial}(X^N)<H+\varepsilon$
  - Even if the probability of error δ is extremely small, **the number of bits per symbol** $\frac{1}{N}H_{\partial}(X^N)$ needed to specify a long N-symbol string **x** with vanishingly small error probability **does not have to exceed H + ε bits**.
  - We need to have only a **tiny tolerance for error**, and the **number of bits** required **drops** significantly from $H_0(X)$ to $(H + \varepsilon)$.

Two parts of Shannon's source coding theorem

- The source coding theorem has two parts !

$$\left|\frac{1}{N}H_\partial(X^N) - H\right| < \varepsilon$$

- $\frac{1}{N}H_\partial(X^N) > H - \varepsilon$ $\frac{1}{N}H_\partial(X^N)$
  - If we are yet more tolerant to compression errors? Even if δ is very close to 1, so that **errors are made most of the time**, the average number of bits per symbol needed to specify **x** must still be at least $H - \varepsilon$ bits !
  - We need to have only a tiny tolerance for error, and the number of bits required drops significantly from H0(X) to (H + ε).

> Regardless of our specific allowance for error,
> the **number of bits per symbol needed to specify x is *H* bits** !

Asymptotic equipartition?

- it is important **not to think** that the elements of the typical set $T_{N\beta}$ really **do have roughly the same probability as each other**
- They are similar in probability only in the sense that their values of $\log_2 1/P(\mathbf{x})$ are within $2N\beta$ of each other.

Why the typical set?

The best choice of subset for block compression is (by definition) S_δ, not a typical set. So why did we bother introducing the typical set? The answer is, *we can count the typical set.* We know that all its elements have 'almost identical' probability (2^{-NH}), and we know the whole set has probability almost 1, so the typical set must have roughly 2^{NH} elements. Without the help of the typical set (which is very similar to S_δ) it would have been hard to count how many elements there are in S_δ.

Further Reading and Summary

Q&A

Further Reading

- **Recommend Readings**
  - Information Theory, Inference, and Learning Algorithms from David MacKay, 2015, pages 74 - 84.
- **Supplemental readings:**

What you should know

- raw bit content
- Ways for compressing files
- The smallest δ-sufficient subset
- The essential bit content of an ensemble
- Why to compress block of symbols
- The Typical set
- Shannon’s source coding theorem. Its two parts
- H(X) viewed as a compression limit for a source

Further Reading and Summary

Q&A