# Lecture 3:
# Data Compression

# Definitions (1)

**Definition 9.** *Discrete Memoryless Source (DMS). A DMS is a discrete-time random process* $\{X_i : i = 1, 2, \ldots\}$ *over the discrete alphabet* $\mathcal{X}$ *such that the random variables* $X_i$ *are mutually independent and identically distributed with common pmf* $P_X$*.* $\Diamond$

**Definition 10.** *A fixed-to-variable length lossless source code of block length* $n$ *with binary output is formed by an encoding function*

$$f : \mathcal{X}^n \to \mathcal{D}^*$$

*where* $\mathcal{D} = \{0, 1\}$ *and* $\mathcal{D}^*$ *denotes the set of all sequences of any length with elements in* $\mathcal{D}$*, and by a decoding function*

$$g : \mathcal{D}^* \to \mathcal{X}^n$$

*such that, for all* $\mathbf{x} \in \mathcal{X}^n$ *we have* $g(f(\mathbf{x})) = \mathbf{x}$*.* $\Diamond$

# Definitions (2)

- For a given fixed-to-variable length lossless source code $(n, f, g)$, the associated length function is defined as

$$\ell(\mathbf{x}) = \text{number of symbols in the sequence } f(\mathbf{x})$$

- The code average length is given by

$$L_n = \mathbb{E}[\ell(X^n)] = \sum_{\mathbf{x} \in \mathcal{X}^n} P_{X^n}(\mathbf{x}) \ell(\mathbf{x})$$

- The (data compression) rate is the normalized average length

$$R = \frac{1}{n} L_n$$

# AEP and data compression (1)

- Using the Asymptotic Equipartition Property (AEP), we can show that for any $\epsilon > 0$ there exist fixed-to-variable lossless source codes of sufficiently large block length $n$ such that

$$\frac{1}{n} L_n \leq H(X) + \delta'(\epsilon)$$

- We associate to all sequences $\mathbf{x} \in \mathcal{T}_\epsilon^{(n)}(X)$ a unique index of length $\lceil n(H(X) + \delta(\epsilon)) \rceil$ binary symbols. Since $|\mathcal{T}_\epsilon^{(n)}(X)| \leq 2^{n(H(X)+\delta(\epsilon))}$, this unique indexing exists.

- We associate to all sequences $\mathbf{x} \notin \mathcal{T}_\epsilon^{(n)}(X)$ a unique index of length $\lceil n \log |\mathcal{X}| \rceil$ binary symbols.

- Codewords corresponding to typical sequences are prefixed by 0, and codewords corresponding to non-typical sequences are prefixed by 1.

# AEP and data compression (2)

- The resulting average length is given by

$$\begin{aligned} L_n &= \sum_{\mathbf{x}\in\mathcal{T}_\epsilon^{(n)}(X)} P_{X^n}(\mathbf{x})\left(\lceil n(H(X)+\delta(\epsilon))\rceil+1\right) \\ &\quad + \sum_{\mathbf{x}\notin\mathcal{T}_\epsilon^{(n)}(X)} P_{X^n}(\mathbf{x})\left(\lceil n\log|\mathcal{X}|\rceil+1\right) \\ &\leq (n(H(X)+\delta(\epsilon))+2)+(n\log|\mathcal{X}|+2)\,\epsilon \end{aligned}$$

Dividing by $n$ we obtain

$$\frac{1}{n}L_n \leq H(X) + \underbrace{\delta(\epsilon)+\frac{2}{n}+\left(\log|\mathcal{X}|+\frac{2}{n}\right)\epsilon}_{\leq\delta'(\epsilon)\ \text{for suff. large}\ n}$$

# Uniquely decodable and prefix codes (1)



- The $K$-th extension code of a $(n, f, g)$ lossless code is obtained by concatenating $K$ blocks of length $n$ of source symbols, encoding them separately, and forming the extension codeword as the concatenation of the codewords. The $K$-th extension encoding function is

$$f^{(K)} : \mathcal{X}^{nK} \to \mathcal{D}^*$$

such that

$$f^{(K)}(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_K) = (f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_K))$$

- The $*$-extension code of a $(n, f, g)$ lossless code is the collection of all $K$-extension codes, for all $K$ (it is a mapping from $\mathcal{X}^*$ to $\mathcal{D}^*$).

# Uniquely decodable and prefix codes (2)



- We are interested in decoding an extension codeword *sequentially*. Since source $n$-blocks yield different length codewords, the sequential decoder needs to be able to identify when a codeword ends and the next starts. Notice: insertion commas in between the codewords (i.e., parsing the output string) requires the use of extra symbols, and therefore increases the overall coding length!

- An $(n, f, g)$ code is called uniquely decodable if its $*$-extension code is one-to-one.

- An $(n, f, g)$ code is called instantaneously decodable (or prefix code) if no codeword is a prefix of another codeword.

- Clearly, the $*$-extension code of a prefix codes is sequentially decodable: the decoder reads the string $(f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_K))$ sequentially, and as soon as a codeword is recognized, then the decoding function $g$ is applied, recovering the corresponding source block.

# Kraft inequality

**Theorem 6.** *The length function $\ell$ of any prefix code $f : \mathcal{X}^n \to \mathcal{D}^*$ satisfies*

$$\sum_{\mathbf{x}\in\mathcal{X}^n} 2^{-\ell(\mathbf{x})} \leq 1$$

*Conversely, given a set of integers $\ell_1, \ell_2, \ldots, \ell_{|\mathcal{X}|^n}$ satisfying this inequality there exists a prefix code with codewords of these lengths.* □

**Proof:** Construct a binary tree of depth $\ell_{\max}$ and then prune the tree such that each codeword $i$ correspond to a leaf at depth $\ell_i$, root of a subtree with $2^{\ell_{\max}-\ell_i}$ leaves. Notice that because of the prefix condition all such subtrees must be disjoint. Therefore, summing over all the leaves of these subtrees we have

$$2^{\ell_{\max}} \geq \sum_i 2^{\ell_{\max}-\ell_i}$$

Dividing both sides by $2^{\ell_{\max}}$ we have the result. ■

- An alternative proof that holds also when the set of lengths is countably infinite consists of taking each binary codeword $(b_1, b_2, \ldots, b_{\ell_i})$ and associate it to the interval

$$\left[\sum_{j=1}^{\ell_i} b_j 2^{-j}, \sum_{j=1}^{\ell_i} b_j 2^{-j} + 2^{-\ell_i}\right)$$

of all real numbers whose binary representation has the first $\ell_i$ significant digits equal to $b_1, \ldots, b_{\ell_i}$.

- By the prefix condition these intervals are disjoint, and their total measure cannot be larger than 1.

# Converse of data compression

**Theorem 7.** *The average normalized length of any $(n, f, g)$ prefix code must satisfy*

$$\frac{1}{n} L_n \geq H(X)$$

□

**Proof:**

We have

$$\begin{aligned} L_n - nH(X) &= \sum_{\mathbf{x} \in \mathcal{X}^n} P_{X^n}(\mathbf{x}) \ell(\mathbf{x}) + \sum_{\mathbf{x} \in \mathcal{X}^n} P_{X^n}(\mathbf{x}) \log P_{X^n}(\mathbf{x}) \\ &= -\sum_{\mathbf{x} \in \mathcal{X}^n} P_{X^n}(\mathbf{x}) \log 2^{-\ell(\mathbf{x})} + \sum_{\mathbf{x} \in \mathcal{X}^n} P_{X^n}(\mathbf{x}) \log P_{X^n}(\mathbf{x}) \end{aligned}$$

$$
\begin{aligned}
&= -\sum_{\mathbf{x}\in\mathcal{X}^n} P_{X^n}(\mathbf{x}) \log\left(\frac{2^{-\ell(\mathbf{x})}}{\sum_{\mathbf{x}'\in\mathcal{X}^n} 2^{-\ell(\mathbf{x}')}}\right) \\
&\quad + \sum_{\mathbf{x}\in\mathcal{X}^n} P_{X^n}(\mathbf{x}) \log P_{X^n}(\mathbf{x}) - \log\left(\sum_{\mathbf{x}'\in\mathcal{X}^n} 2^{-\ell(\mathbf{x}')}\right) \\
&= D\left(P_{X^n}\|Q_{X^n}\right) - \log\left(\sum_{\mathbf{x}'\in\mathcal{X}^n} 2^{-\ell(\mathbf{x}')}\right) \\
&\geq D\left(P_{X^n}\|Q_{X^n}\right) \\
&\geq 0
\end{aligned}
$$

where we used the Kraft inequality to establish that $\log\left(\sum_{\mathbf{x}'\in\mathcal{X}^n} 2^{-\ell(\mathbf{x}')}\right) \leq 0$, and we define $Q_{X^n}(\mathbf{x}) = \frac{2^{-\ell(\mathbf{x})}}{\sum_{\mathbf{x}'\in\mathcal{X}^n} 2^{-\ell(\mathbf{x}')}}$. ■

# Kraft inequality for uniquely decodable codes



- Kraft inequality holds also for uniquely decodable codes (see Cover-Thomas, Theorem 5.5.1 and Corollary at page 117).
- This implies that even by relaxing the prefix requirement the same lower bound on the optimal coding length holds.
- The good news are that there is no loss of optimality and generality by restricting to prefix codes.
- Clearly prefix codes have the advantage that they are “instantaneous”: while scanning a sequence of codewords, as soon as a codeword is recognized the decoder can immediately output the corresponding message.

# Bounds on the optimal length

- From the proof of the previous theorem we have that

$$L_n - nH(X) = D\left(P_{X^n} \| Q_{X^n}\right) - \log\left(\sum_{\mathbf{x}' \in \mathcal{X}^n} 2^{-\ell(\mathbf{x}')}\right) \geq 0$$

- If the quantities $-\log P_{X^n}(\mathbf{x})$ are integer for each $\mathbf{x} \in \mathcal{X}^n$, the length function

$$\ell(\mathbf{x}) = -\log P_{X^n}(\mathbf{x})$$

yields $L_n = nH(X)$ and

$$\sum_{\mathbf{x} \in \mathcal{X}^n} 2^{-\ell(\mathbf{x})} = \sum_{\mathbf{x} \in \mathcal{X}^n} 2^{\log P_{X^n}(\mathbf{x})} = \sum_{\mathbf{x} \in \mathcal{X}^n} P_{X^n}(\mathbf{x}) = 1$$

- These optimal lengths, however, may not be possible since in general $-\log P_{X^n}(\mathbf{x})$ is not an integer.
- Hence, we choose

$$\ell(\mathbf{x}) = \lceil -\log P_{X^n}(\mathbf{x}) \rceil \leq -\log P_{X^n}(\mathbf{x}) + 1$$

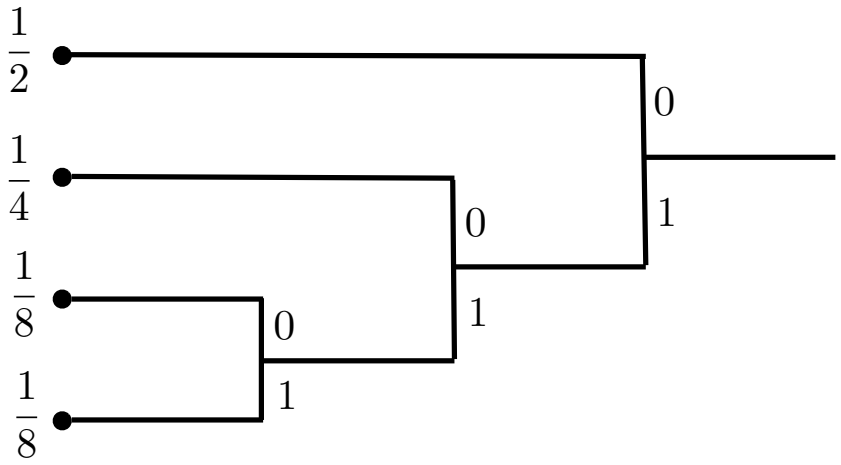
- This choice yields

$$L_n \leq nH(X) + 1$$

- Putting everything together, we have shown that there exist prefix codes with normalized average length satisfying

$$H(X) \leq \frac{1}{n} L_n \leq H(X) + \frac{1}{n}$$

# Huffman coding




| | |
|---|---|
| a | 0 |
| b | 10 |
| c | 110 |
| d | 111 |

# Huffman coding: Example



- $\mathcal{X} = \{1, 2, 3, 4, 5\}$, $P_X \equiv (0.25, 0.25, 0.2, 0.15, 0.15)$, $n = 1$.

# Huffman coding: Example



- $\mathcal{X} = \{1, 2, 3\}$, $P_X \equiv (0.5, 0.3, 0.2)$, $n = 2$.

# Huffman codes and Shannon codes



- We refer to prefix codes that use lengths $\ell(\mathbf{x}) = \lceil -\log P_{X^n}(\mathbf{x}) \rceil$ as "Shannon codes".

- The length of a Shannon code may be much worse than the optimal length for some symbol.

**Example:** Consider a binary alphabet $\mathcal{X} = \{0, 1\}$ with $P_X(0) = 0.9999$ and $P_X(1) = 0.0001$. The (optimal) Huffman coding length for $n = 1$ are $\ell(0) = \ell(1) = 1$, while the Shannon coding lengths are $\ell(0) = 1$ and $\ell(1) = 14$.

However, their average length $L_n$ is in between $nH(X)$ and $nH(X) + 1$ for both codes, and therefore it differs by no more than 1 bit. $\Diamond$

# Optimality of Huffman codes (1)



- We let $|\mathcal{X}|^n = M$, and order the probability vector $\mathbf{p}$ corresponding to the pmf $P_{X^n}$ such that $p_1 \geq p_2 \geq \cdots \geq p_M$. Correspondingly, we let $\ell_i$ denote the $i$-th length.

- An optimal code $f$ is a mapping with lengths $\{\ell_i\}$ such that $\sum_{i=1}^{M} p_i \ell_i$ is minimum.

**Lemma 9.** *For given $\mathbf{p}$ there exists an optimal code satisfying the following properties:*

1. *If $p_i \geq p_j$ then $\ell_i \leq \ell_j$.*

2. *The two longest codewords have the same length.*

3. *Two of the longest codewords differ only in the last bit and correspond to symbols $M-1$ and $M$ (the two symbols with smallest probabilities).* □

**Proof:**

It follows by constructing the binary tree of the codewords and swapping (property 1), trimming (property 2), and rearranging (property 3), after noticing that in an optimal code any maximal length codeword must have a "sibling". ■

- Summarizing: for $\mathbf{p}$ with $p_1 \geq p_2 \geq \cdots \geq p_M$ there exists an optimal code with $\ell_1 \leq \ell_2 \leq \cdots \leq \ell_{M-1} = \ell_M$, and where $f(M-1)$ and $f(M)$ differ only by the last bit.

- Such optimal code is referred to as a canonical code.

# Optimality of Huffman codes (3)



- Huffman reduction procedure: from $\mathbf{p}$ of $M$ elements, by merging the two smallest probabilities $p_{M-1}$ and $p_M$, we obtain $\mathbf{p}'$ of $M-1$ elements.
- Let $f'$ denote an optimal canonical code for $\mathbf{p}'$, with lengths $\{\ell'_i\}$.
- We expand $f'$ and construct a code $f$ for $\mathbf{p}$ as follows: append symbol 0 to $f'(M-1)$ to form $f(M-1)$, and append symbol 1 to $f'(M-1)$ to form $f(M)$. For all other $1 \leq i \leq M-2$ let $f(i) = f'(i)$.
- We have

$$L = \sum_{i=1}^{M} p_i \ell_i = \sum_{i=1}^{M-2} p_i \ell'_i + p_{M-1}(\ell'_{M-1}+1) + p_M(\ell'_{M-1}+1) = L^\star(\mathbf{p}') + p_{M-1} + p_M$$

# Optimality of Huffman codes (4)



- Let $f$ denote an optimal canonical code for $\mathbf{p}$, with lengths $\{\ell_i\}$.
- We reduce $f$ and construct a code $f'$ for $\mathbf{p}'$ as follows: merge the last two symbols $M-1$ and $M$, and use $\ell'_{M-1} = \ell_M - 1$. For all other $1 \leq i \leq M-2$ let $f'(i) = f(i)$.
- We have

$$L' = \sum_{i=1}^{M-1} p'_i \ell'_i = \sum_{i=1}^{M-2} p_i \ell_i + (p_{M-1} + p_M)(\ell_M - 1) = L^\star(\mathbf{p}) - p_{M-1} - p_M$$

# Optimality of Huffman codes (5)



- Adding these relationships together and rearranging terms we obtain

$$(L - L^\star(\mathbf{p})) + (L' - L^\star(\mathbf{p}')) = 0$$

- Since by assumption $L - L^\star(\mathbf{p}) \geq 0$ and $L' - L^\star(\mathbf{p}') \geq 0$, we conclude that $L = L^\star(\mathbf{p})$ and that $L' = L^\star(\mathbf{p}') = 0$.

- We have shown that by extending an optimal code for $\mathbf{p}'$ we obtain an optimal code for $\mathbf{p}$, and by reducing an optimal code for $\mathbf{p}$ we obtain an optimal code for $\mathbf{p}'$.

# Optimality of Huffman codes (6)

**Theorem 8.** *Huffman coding is optimal, i.e., for given $P_{X^n}$ the Huffman procedure constructs a $(n, f, g)$ prefix code with minimal $L_n$.* □

**Proof:**

We operate by induction on the alphabet size. For alphabets of size 2, Huffman coding yields $\ell_1 = \ell_2 = 1$, which is clearly optimal. Assuming that we have an optimal code for alphabet size $M - 1$, we construct an optimal code for alphabet size $M$ by extension. Then, we notice that if we start from an alphabet of size $M$ and apply Huffman coding construction, we proceed step-by-step recovering the same sequence of optimal codes that we have constructed before. It follows that Huffman coding is optimal for any alphabet size $M$.

■

# Shannon-Fano-Elias coding (1)

- Idea: associate the source vectors $\mathbf{x} \in \mathcal{X}^n$ to points on the unit interval $[0, 1]$, and then use truncation of the binary expansion of these points as codewords.

- We order the vectors $\mathbf{x} \in \mathcal{X}^n$ in lexicographic order, and associate them to indices $i = 1, 2, \ldots, |\mathcal{X}|^n = M$, with the corresponding $P_{X^n}(\mathbf{x}) = p_i$.

- Define the two functions

$$F(i) = \sum_{j \leq i} p_j \quad \text{and} \quad \overline{F}(i) = \sum_{j < i} p_j + \frac{p_i}{2}$$

- Notice that $\overline{F}(i)$ is the middle point of the interval $[F(i-1), F(i)]$, and that $F(0) = 0$ and $F(M) = 1$.

- The real numbers $\overline{F}(i)$ uniquely identify the $i$-th source vectors.

# Shannon-Fano-Elias coding (2)



- We use a *truncation* of these real numbers, represented in binary form: using $\ell_i = \lceil -\log p_i \rceil + 1$, the round-off error is

$$\overline{F}(i) - \lfloor \overline{F}(i) \rfloor_{\ell_i} \leq 2^{-\ell_i} \leq \frac{p_i}{2} = \overline{F}(i) - F(i-1)$$

- Hence, the truncated value $\lfloor \overline{F}(i) \rfloor_{\ell_i}$ belongs to the interval $[F(i-1), F(i)]$ and it can be used for uniquely representing the $i$-th source vector.

# Shannon-Fano-Elias coding (2)



- Prefix condition: let $0.b_1 b_2 \ldots b_{\ell_i}$ be the binary expansion of the point $\lfloor \overline{F}(i) \rfloor_{\ell_i}$. Codewords satisfy the prefix condition is the intervals

$$\left[0.b_1 b_2 \ldots b_{\ell_i}, \;\; 0.b_1 b_2 \ldots b_{\ell_i} + 2^{-\ell_i}\right)$$

  are disjoint for all $i$.

- We know that this condition is satisfied, since this interval is contained in $[F(i-1), F(i))$, for all $i$.

- Average length:

$$L_n = \sum_{i=1}^{M} p_i \ell_i = \sum_{\mathbf{x} \in \mathcal{X}^n} P_{X^n}(\mathbf{x}) \left( \lceil -\log P_{X^n}(\mathbf{x}) \rceil + 1 \right) \leq nH(X) + 2$$

# Arithmetic coding

- As for Huffman coding, constructing the function $\overline{F}(i)$ for large $n$ is impractical (exponentially large number of points).
- Fortunately, the CDF $F(i)$ and therefore the function $\overline{F}(i)$ can be constructed sequentially, step by step, by considering the sequence of symbols $x_1, x_2, \ldots, x_n$.
- This sequential procedure, together with specific implementations that require only finite-precision arithmetic, forms the basis for Arithmetic Coding.
- Arithmetic Coding is currently used in a number of standard compression algorithms, in particular, to compress sequences of quantization symbols in lossy source coding schemes (e.g., image coding JPEG/JPEG2000).
- A glimpse on correlated processes: when the source sequence $X^n$ is a Markov chain, Arithmetic still works if we know the sequence of conditional pmfs $P_{X_i|X_{i-1}}(x_i|x_{i-1})$, for $i = 1, \ldots, n$.

**Theorem 9.** *Let $X^n \sim P_{X^n}$ and let $Q_{X^n}$ be another product pmf defined on $\mathcal{X}^n$. The average length of a code for $X^n$ with lengths $\ell(\mathbf{x}) = \lceil -\log Q_{X^n}(\mathbf{x}) \rceil$ satisfies:*

$$H(X) + D(P_X \| Q_X) \leq \frac{1}{n} L_n \leq H(X) + D(P_X \| Q_X) + \frac{1}{n}$$

□

**Proof:**

We have

$$\begin{aligned} L_n &= \sum_{\mathbf{x}\in\mathcal{X}^n} P_{X^n}(\mathbf{x}) \left\lceil -\log Q_{X^n}(\mathbf{x}) \right\rceil \\ &\leq \sum_{\mathbf{x}\in\mathcal{X}^n} P_{X^n}(\mathbf{x}) \left( -\log Q_{X^n}(\mathbf{x}) + 1 \right) \\ &= \sum_{\mathbf{x}\in\mathcal{X}^n} P_{X^n}(\mathbf{x}) \log \frac{P_{X^n}(\mathbf{x})}{Q_{X^n}(\mathbf{x})} + \sum_{\mathbf{x}\in\mathcal{X}^n} P_{X^n}(\mathbf{x}) \log \frac{1}{P_{X^n}(\mathbf{x})} + 1 \\ &= nD(P_X \| Q_X) + nH(X) + 1 \end{aligned}$$

■

# Competitive optimality of Shannon codes



- Game: two people A and B are given a pmf $P_{X^n}$ and are asked to design codes for this probability distribution.

- A source vector $X^n$ is generated according to $P_{X^n}$ and player A has payoff $+1$, $-1$ or $0$ if $\ell^{(A)}(X^n) < \ell^{(B)}(X^n)$, $\ell^{(A)}(X^n) > \ell^{(B)}(X^n)$ or $\ell^{(A)}(X^n) = \ell^{(B)}(X^n)$, respectively.

- The next result show that the length of a Shannon code cannot be much worse than the length of any other uniquely decodable code, with high probability.

**Theorem 10.** *Let $X^n \sim P_{X^n}$, and let $\ell(\mathbf{x}) = \lceil -\log P_{X^n}(\mathbf{x}) \rceil$ denote the length function of a Shannon code. For any other uniquely decodable code with length function $\ell'(\mathbf{x})$, we have*

$$\mathbb{P}\left(\ell(X^n) \geq \ell'(X^n) + c\right) \leq \frac{1}{2^{c-1}}$$

□

## Proof:

$$\begin{aligned}
\mathbb{P}\left(\ell(X^n) \geq \ell'(X^n) + c\right) &= \mathbb{P}\left(\lceil -\log P_{X^n}(X^n) \rceil \geq \ell'(X^n) + c\right) \\
&\leq \mathbb{P}\left(-\log P_{X^n}(X^n) \geq \ell'(X^n) + c - 1\right) \\
&= \mathbb{P}\left(P_{X^n}(X^n) \leq 2^{-\ell'(X^n)-c+1}\right) \\
&= \sum_{\mathbf{x}: P_{X^n}(\mathbf{x}) \leq 2^{-\ell'(\mathbf{x})-c+1}} P_{X^n}(\mathbf{x}) \\
&\leq \sum_{\mathbf{x}: P_{X^n}(\mathbf{x}) \leq 2^{-\ell'(\mathbf{x})-c+1}} 2^{-\ell'(\mathbf{x})-c+1} \\
&= 2^{-c+1} \sum_{\mathbf{x}: P_{X^n}(\mathbf{x}) \leq 2^{-\ell'(\mathbf{x})-c+1}} 2^{-\ell'(\mathbf{x})} \\
&\leq 2^{-c+1}
\end{aligned}$$

where the last step follows from Kraft inequality. ■

# End of Lecture 3