3. Data Compression, AEP and Lossless Source Coding

Vaughan Sohn

October 7, 2024

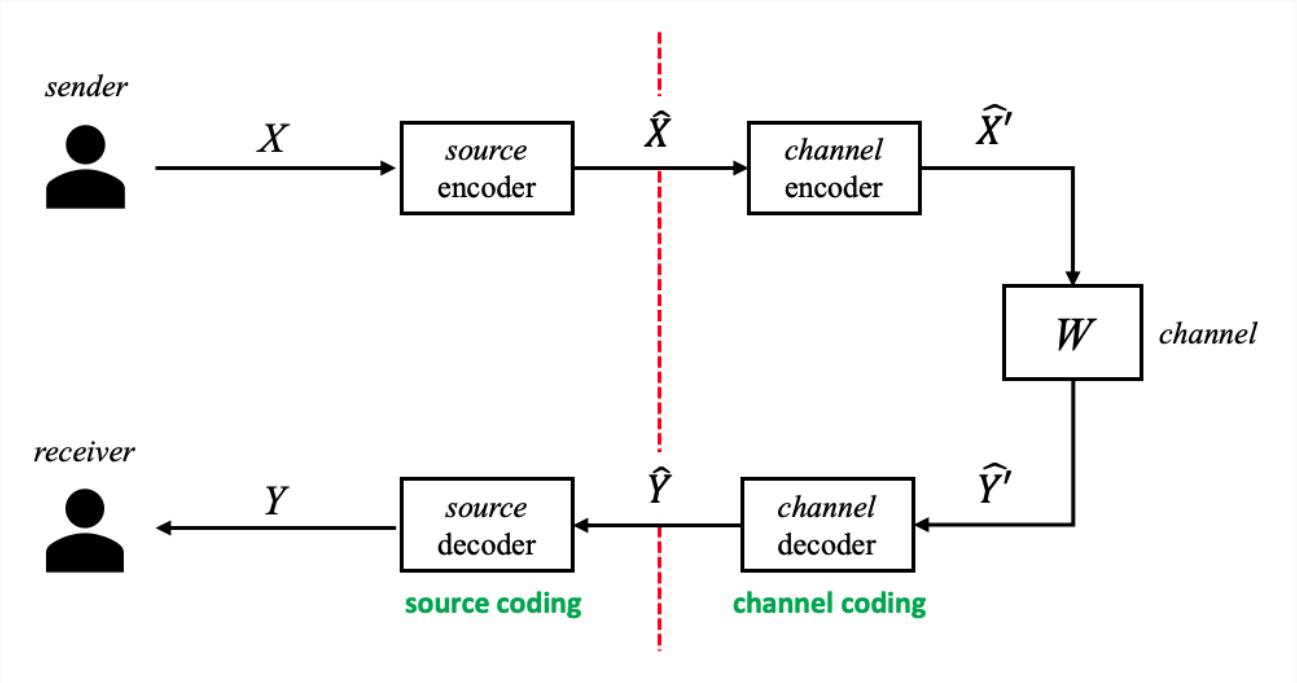
Contents

Source coding

Communication System

Communication system을 표현할 때, 정보이론에서는 "Shannon"이 제안한 *digital communication*을 따른다.

- Shannon의 아이디어는 source와 channel을 나타낼 때, 일관되는 하나의 양식인 binary representation을 따르게 하는 것이다.
- Interface를 이용하면, source data의 종류나 channel의 물리적 특성에 관계없이 encoder/decoder만 잘 설계하면 동일한 framework를 적용할 수 있다!
- 이번 챕터에서는 "Source coding"에 대해 다루고자 한다.



Source coding

Source coding은 크게 2가지로 이루어져 있다.

- source encoder: source data를 bit string으로 encoding한다.
- source decoder: encoded data를 다시 source data로 decoding한다.

Question

어떻게 해야 "좋은 encoder"를 설계할 수 있는가?

Source data가 다음의 특성을 가진다고 가정하자.

- source는 특정 alphabet $\mathcal{X}$에 속한다.
- source sequence는 i.i.d이다.
- source는 특정 distribution P를 따른다.

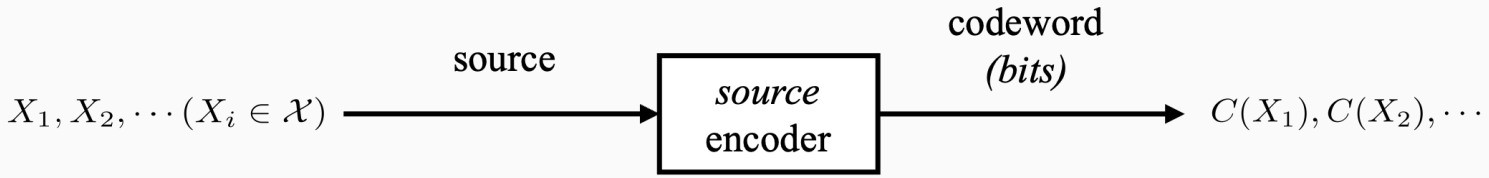
Definition 1 (Discrete Memoryless Source: DMS)

A sequence $\{X_i\}_{i=1}^{\infty}$ $\sin i.i.d P$ on $\mathcal{X}$ is called a **DMS(P)**

Fixed-length source coding

Definition 2 (fixed-length code)

A **Fixed-length code** is a code where each codeword $C(x)$ is restricted to have the *same block length* L.



$$C : \mathcal{X} \to \{0,1\}^L$$

- *Decodability*를 위하여 fixed-length code의 length는 다음 조건을 만족해야한다.

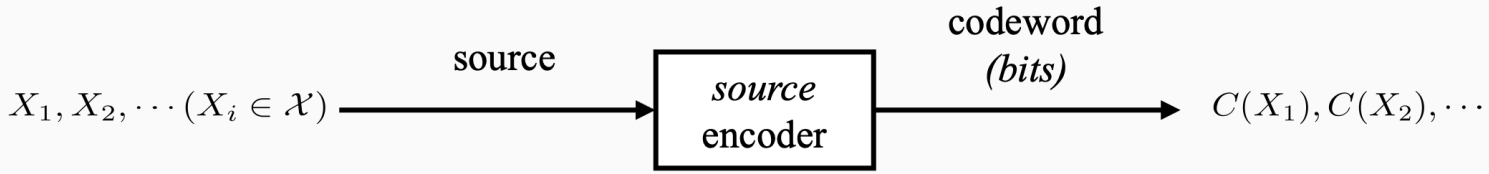
$$\log M \leq L < \log M + 1, \qquad (M = |\mathcal{X}|)$$

- Example:
  If the alphabet $\mathcal{X}$ consists of the 7 symbols $\{a, b, c, d, e, f, g\}$,

Variable-length source coding

Definition 3 (variable-length code)

A **Variable-length code** is a code where each codeword $C(x)$ can have a different length; number of bits $l(x)$ of $C(x)$.



$$C : \mathcal{X} \to \{0,1\}^*$$

- Variable-length code의 length 대한 기댓값 R은 다음과 같이 계산할 수 있다.

$$R = \sum_{x \in \mathcal{X}} p(x) \cdot l(x)$$

- (intuitive) 더 낮은 빈도로 나타나는 symbol에 더 긴 길이의 codeword를 할당하면, expected length of codeword R을 줄일 수 있을 것이다.
- (problem) 그러나, variable-length code는 fixed-length code와 다르게 어디까지가 하나의 codeword인지를 판단하기 어렵다.
  $\rightarrow$*Issue of decodability!*

Decodability and optimality

Uniquely decodable code

Definition 4 (extension)

The **extension** C^* of a code C is the mapping from finite-length string of $\mathcal{X}$ to finite-length string of $\mathcal{D} = \{0, 1\}$, defined by

$$C(x_1 x_2 \cdots x_n) = C(x_1)C(x_2) \cdots C(x_n).$$

Definition 5 (unique decodability)

A code is called **uniquely decodable** if its extension is *non-singular*. In other words, any encoded string in a uniquely decodable code has only one possible source string producing it.

✓meaning: extension은 주어진 source data sequence를 codeword sequence로 나타낸 data이다. 만약 이 extension으로부터 다시 원본 source data를 복원할 수 있다면, 그 code는 uniquely decodable이다.

Prefix-free code

Definition 6 (prefix-free code)

A code is called a **prefix-free code** if no codeword is a *prefix* of any other codeword.

$$C(x_j) \notin \{C_1(x_i), C_2(x_i), \cdots, C_n(x_i)\}, \qquad (\forall i, j, (i \neq j))$$

where $x_i = b_1 b_2 \cdots b_n$ and $C_k(x_i) = b_1 b_2 \cdots b_k$.

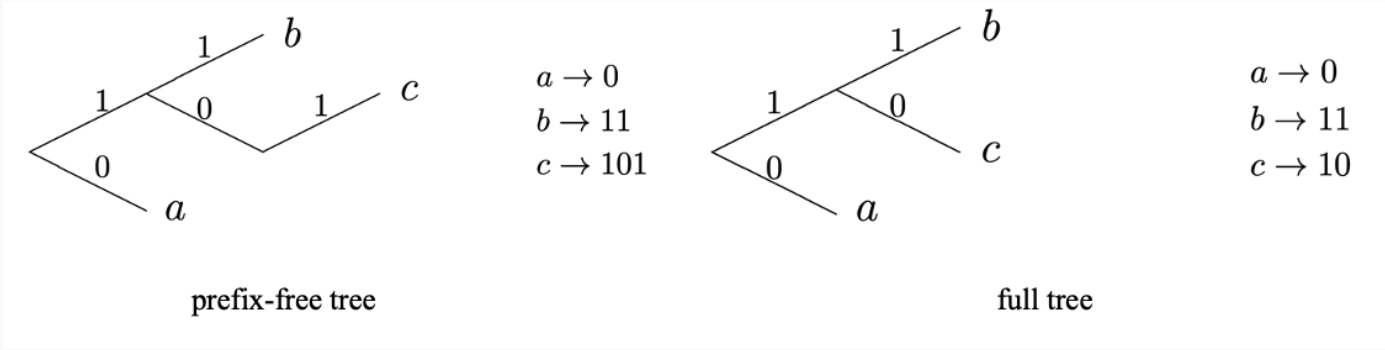
- prefix는 어떤 bit string의 initial substring을 의미한다.
- prefix-free code는 어떤 codeword의 prefix도 *자기 자신이 아닌 다른 codeword가 되지 않도록* 설계한 code를 의미한다.
- Example:

$$\begin{aligned} C(a) &= 0 && \rightarrow (\text{prefix: } 0) \\ C(b) &= 10 && \rightarrow (\text{prefix: } \{1, 10\}) \\ C(c) &= 11 && \rightarrow (\text{prefix: } \{1, 11\}) \end{aligned}$$

Binary-tree representation

Source code의 encoding scheme는 binary tree로 표현할 수 있다.

- prefix-free code를 tree로 나타내면, 모든 codeword가 leaf node에 위치하게 된다.
- full tree는 prefix-free 구조를 깨지 않고서는 더이상 새로운 노드를 추가할 수 없는 형태의 tree를 의미한다.
- 아래와 같이 $|\mathcal{X}| = 3$에 대해 full tree가 되도록 code를 설계하면, tree의 depth가 줄어들기에 codeword 길이의 기댓값이 줄어든다.



Theorem 7

Every prefix-code is uniquely decodable.

$$\textit{Prefix-free} \subset \textit{Uniquely decodable} \subset \textit{code}$$

$\Rightarrow$ Prefix-code의 parsing은 binary-tree에 대한 searching을 *leaf node를 만날 때까지* 수행하면 되기 때문에 자명하게 uniquely decodable이다.□

Condition of optimal code

Lemma 8

Optimal codes have the property that if $p_i > p_j$, then $l_i \leq l_j$.

$*$ Proof: (귀류법) If $p_i > p_j$, then $l_i > l_j$라고 가정하자.
$\Rightarrow$

Lemma 9

Optimal prefix-free codes have the property that the associated code tree is full.

$*$ Proof: 만약 full tree가 아니라면, 언제나 tree depth를 더 줄일 수 있는 다른 full tree를 찾아낼 수 있기 때문에 optimal code가 될 수 없다.□

Lemma 10

Two symbols with min. probability have the same length. In other words, If $p_1 \geq p_2 \geq \cdots \geq p_{M-1} \geq p_M$, then $l_{M-1} = l_M$.

$*$ Proof:
By Lemma 8, if $p_1 \geq p_2 \geq \cdots \geq p_{M-1} \geq p_M$, then $l_1 \leq l_2 \leq \cdots \leq l_{M-1} \leq l_M$.
By Lemma 9, full tree여야 하기 때문에, 가장 확률이 작은 두 노드의 depth는 동일해야한다. $\rightarrow$ sibling node □

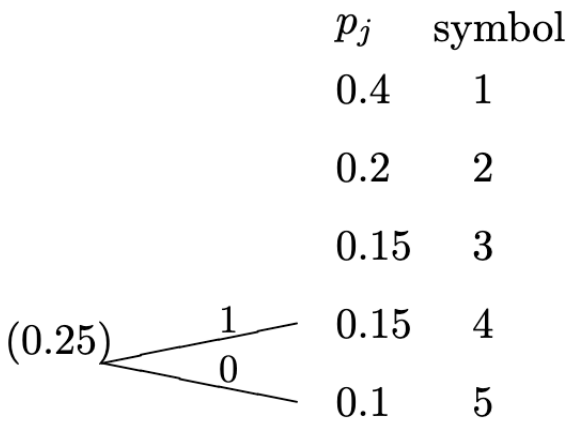
Huffman Code

Huffman Code Algorithm

For $X \in \mathcal{X}$, X be a random symbol with pmf $p_1 \geq p_2 \geq \cdots \geq p_{M-1} \geq p_M$,

1. Choose two least likely symbols (p_M, p_{M-1}) and constraining them to be siblings.
2. Now we consider *new data compression problem* with pmf $\{p_1, p_2, \cdots, p_{M-2}, p_M + p_{M+1}\}$
3. Repeat step 1 & 2 until there is no more remaining symbol.

Example:



The two least likely symbols, 4 and 5 have been combined as siblings. The reduced set of probabilities then becomes {0.4, 0.2, 0.15, 0.25}.

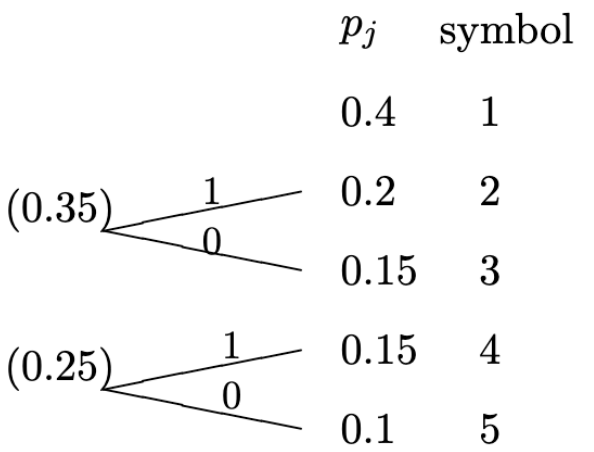
Figure 2.9: Step 1 of the Huffman algorithm; finding X' from X

Huffman Code Algorithm

For $X \in \mathcal{X}$, X be a random symbol with pmf $p_1 \geq p_2 \geq \cdots \geq p_{M-1} \geq p_M$,

1. Choose two least likely symbols (p_M, p_{M-1}) and constraining them to be siblings.
2. Now we consider *new data compression problem* with pmf $\{p_1, p_2, \cdots, p_{M-2}, p_M + p_{M+1}\}$
3. Repeat step 1 & 2 until there is no more remaining symbol.

Example (contd.):



The two least likely symbols in the reduced set, with probabilities 0.15 and 0.2, have been combined as siblings. The reduced set of probabilities then becomes {0.4, 0.35, 0.25}.

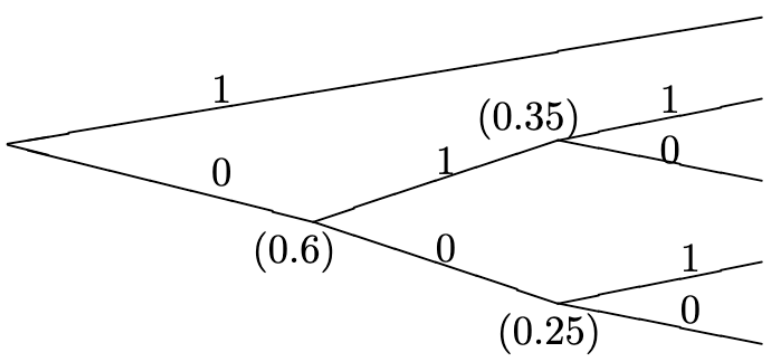
Figure 2.10: Finding X'' from X'.

Huffman Code Algorithm

For $X \in \mathcal{X}$, X be a random symbol with pmf $p_1 \geq p_2 \geq \cdots \geq p_{M-1} \geq p_M$,

1. Choose two least likely symbols (p_M, p_{M-1}) and constraining them to be siblings.
2. Now we consider *new data compression problem* with pmf $\{p_1, p_2, \cdots, p_{M-2}, p_M + p_{M+1}\}$
3. Repeat step 1 & 2 until there is no more remaining symbol.

Example (contd.):



p_j	symbol	codeword
0.4	1	1
0.2	2	011
0.15	3	010
0.15	4	001
0.1	5	000

Figure 2.11: The completed Huffman code.

Optimality of Huffman code

Lemma 11

Huffman algorithm constructs the **optimal** *prefix-free code.* $\rightarrow$ *minimum* R.

$*$ Proof: X에 대한 average length를 $X' = X - \{M-1, M\}\}$에 대한 average length로 표현한 식은 다음과 같다.

$$R = R' + p_{M-1} + p_M$$

각 step마다 subproblem R'을 minimize하기 때문에, 다음을 이끌어낸다.

$$R_{min} = R'_{min} + p_{M-1} + p_M$$

따라서 Huffman algorithm은 optimal이다.□

Asmptotic Equipartition Property (AEP)

Prerequisites: Weak Law of Large Numbers

Theorem 12 (Weak Law of Large Numbers (WLLN))

Let $X_1, X_2, \cdots$ be i.i.d with mean μ and variance $\sigma^2 < \infty$. Let define a new random variable S_n as

$$S_n \triangleq \frac{1}{n}(X_1 + X_2 + \cdots + X_n).$$

Then, if n is increasing then S_n convergence to μ.

$$S_n \xrightarrow{P} \mu, \quad , \text{ i.e., } \quad (S_n - \mu) \xrightarrow{P} 0.$$

In other words, for all $\epsilon, \delta > 0$, there exists N_0 such that $\forall n > N_0$,

$$Pr(|S_n - \mu| > \delta) < \epsilon,$$

i.e., for all $\delta > 0$,

$$\lim_{n \to \infty} Pr[|S_n - \mu| > \delta] = 0.$$

✓meaning: Sample의 개수 n이 증가할 수록, S_n의 값이 X_i의 기댓값에 가까워진다는 내용을 다룬다. n이 커질수록 variance는 0에 수렴하기 때문에 점점 분포가 sharp해진다.

$$\mathbb{E}[S_n] = \mu, \quad Var[S_n] = \frac{\sigma^2}{n}$$

Prerequisites: Weak Law of Large Numbers

For proof of WLLN, we using markov inequality.

Theorem 13 (Markov inequality)

If $Z \geq 0$ *and* $\gamma > 0$, *then*

$$Pr(Z > \gamma) < \frac{\mathbb{E}[Z]}{\gamma}.$$

✓meaning: $[0, \infty)$ 범위에 있는 random variable Z에 대해서, Z가 특정 값 γ보다 클 확률은 γ에 대한 $\mathbb{E}[Z]$의 비율을 상한으로 가진다.

∗ Proof of theorem 13: (hint) $\mathbb{E}[Z]$를 2개의 event에 대해 decomposition하자.

$$\mathbb{E}[Z] = \underbrace{\mathbb{E}[Z|Z > \gamma]}\Pr(Z > \gamma) + \underbrace{\mathbb{E}[Z|Z \leq \gamma]}\Pr(Z \leq \gamma).$$

⇒

∗ Proof of theorem 12: (hint) R.v. $Z \triangleq |S_n - \mu|^2$를 정의하자.

⇒

Asymptotic Equipartition Property (AEP)

Theorem 14 (Asymptotic Equipartition Property (AEP))

Let $X_1, X_2, \cdots, X_n$ *be i.i.d with* P *over* $\mathcal{X}$. *Consider* $\boxed{Y_i = -\log P(X_i)}$. *Then, by definition of entropy* $\mathbb{E}[Y_i] = H(P)$ *(*). Let us denote* $(x_1, x_2, \cdots, x_n)$ *by* x_1^n *or* x^n. *By applying WLLN,* $\forall \epsilon, \delta > 0$, $\exists N_0$ *such that* $\forall n > N_0(\epsilon, \delta)$,

$$P^n\left[\left\{x_1^n : \left|\underbrace{-\frac{1}{n}\sum_{i=1}^{n}\log P(x_i)}_{\text{empirical mean}} - \underbrace{H(P)}_{\text{expectation}}\right| > \delta\right\}\right] < \epsilon$$

✓meaning: Empirical mean과 expectation의 차이가 δ보다 크게 만드는 sequence x_1^n들의 집합의 확률은 ϵ보다 작다.

- P^n은 확률분포 P를 n번 곱한 것으로, i.i.d이기 때문에 joint probability가 각 marginal probability들의 곱과 동일하기 때문이다.
- $\log$함수의 성질에 의하면, summation of log를 다음과 같이 바꾸어 쓸 수 있다.

$$P^n\left[\left\{x_1^n : \left|-\frac{1}{n}\log P^n(x_1^n) - H(P)\right| > \delta\right\}\right] < \epsilon$$

Weak typical set

Definition 15 (weak typical set)

We define **weak typical set** as,

$$A_\delta^{(n)}(P) \triangleq \left\{ x_1^n \in \mathcal{X}^n : \left| -\frac{1}{n} \log P^n\left(x_1^n\right) - H(P) \right| \leq \delta \right\}.$$

✓meaning: Empirical mean과 expectation의 차이가 δ보다 작게 만드는 sequence x_1^n들의 집합을 weak typical set이라고 정의한다.

Corollary 16

Theorem 14 with typical set $A_\delta^{(n)}(P)$ implies that

$$P^n\left(A_\delta^{(n)}(P)\right) \geq 1 - \epsilon.$$

- Weak typical set도 x의 값에는 영향을 받지않고 그 r.v.가 따르는 확률분포에만 영향받기 때문에 P로 표현한다.
- 기호의 단순화를 위해 $\delta = \epsilon$으로 두자. $\rightarrow A_\epsilon^{(n)}(P)$

Weak typical set

$A_\epsilon^{(n)}(P)$를 이용하면, weak typical set이 가지는 아주 중요한 성질 하나를 보일 수 있다.

- By definition of weak typical set,

$$A_\epsilon^{(n)}(P) \triangleq \left\{ x_1^n \in \mathcal{X}^n : \left| -\frac{1}{n} \log P^n\left(x_1^n\right) - H(P) \right| \leq \epsilon \right\}.$$

- set의 elements의 표현에서 절댓값 기호를 제거한 뒤

$$= \left\{ x_1^n \in \mathcal{X}^n : -\epsilon \leq -\frac{1}{n} \log P^n\left(x_1^n\right) - H(P) \leq \epsilon \right\},$$

- set의 each element x_1^n에 대한 확률에 대해 부등식을 정리하면 다음을 얻는다.

$$= \left\{ x_1^n : 2^{-n(H(P)+\epsilon)} \leq P^n\left(x_1^n\right) \leq 2^{-n(H(P)-\epsilon)} \right\}$$

$\Rightarrow$ 즉, weak typical set안에 들어있는 sequence들은 거의 유사한 확률을 가진다!

Remarks

- 대부분의 sequence들은 weak typical set안에 존재한다.
- typical set안에 있는 sample들의 확률은 거의 동일하다.

Weak typical set

Corollary 17 (Size of typical set)

- *For typical set* $A_\epsilon^{(n)}(P)$,
$$|A_\epsilon^{(n)}(P)| \leq 2^{n(H(P)+\epsilon)}$$
- *For large enough* n,
$$|A_\epsilon^{(n)}(P)| \geq (1-\epsilon)2^{n(H(P)-\epsilon)}$$

Weak typical set

Corollary 18

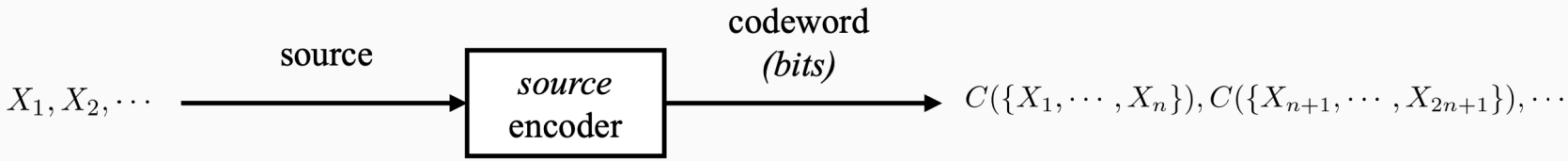
With exponential approximation

Loseless source coding

Fixed-length block coding

Definition 19 (fixed-length code)

A **Fixed-length block code** is a code where a source block x^n , consisting of n symbols from the source alphabet $(x_i \in \mathcal{X})$, is represented by a codeword $C(x^n)$.



$$C : \mathcal{X}^n \to \{0,1\}^*$$

- Block code를 사용했을 때, 각 symbol에 대한 code word length의 기댓값 R_n은 다음과 같이 계산할 수 있다.

$$R_n = \frac{1}{n} \sum_{x^n \in \mathcal{X}^n} p(x^n) \cdot l(x^n) \qquad \text{(bits/source)}$$

References

- T. M. Cover and J. A. Thomas. Elements of Information Theory, Wiley, 2nd ed., 2006.
- Gallager (2008), Principles of Digital Communication, Cambridge University Press.
- Lecture notes for EE623: Information Theory (Fall 2024)