

**תקציר-הרצאות בקורס  
דחיסת נתונים ואותות  
0510-7104**

**סמסטר ב' - 2005**

**מרצה: פרופ' רם זמיר**

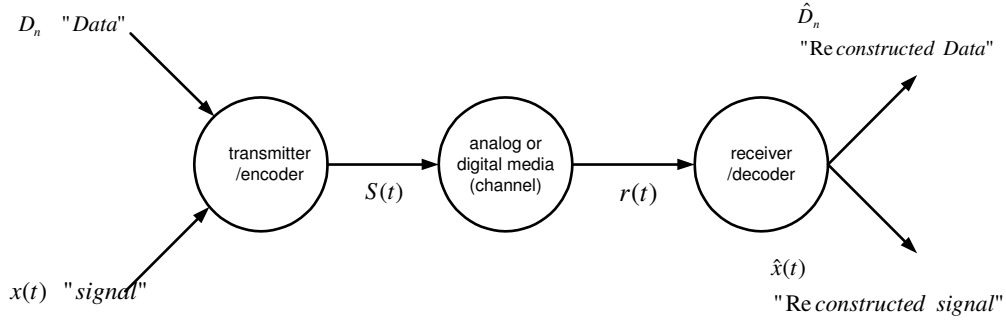
## תוכן עניינים

2	..... התוכן עניינים
3	.....1 - הרצאה
3	..... דחיסת נתונים - מבוא
12	.....2 - הרצאה
12	..... (ARITHMETIC CODING) קידוד אריתמטי
18	.....3 - הרצאה
18	..... UNIVERSAL LOSSLESS SOURCE CODING
28	.....4 - הרצאה
28	..... LEMPEL-ZIV דחיסה של
43	.....5 - הרצאה
43	..... LOSSY COMPRESSION
52	.....6 - הרצאה
52	..... שתי תורות אסימפטוטיות של קוונטיזציה: במימד וברזולוציה
61	.....7 - הרצאה
61	..... ENTROPY CODED DITHERED LATTICE QUANTIZER
71	.....8 - הרצאה
71	..... COMPRESSION UNDER UNCERTAINTY CONDITIONS

# הרצאה מס. 1-דחיסת נתונים

**סוכם ע"י עידו מירון**

תרחיש תקשורת כללי:



בעיות וקריטריונים בתקשורת:

(1) אילוץ שידור וקידוד. לדוגמא: אילוץ הספק  $\frac{1}{T} \int_0^T s(t)^2 dt \leq P$ , אילוץ ספקטרום

שידור וכדומה.

(2) הערכת הביצועים:

הערכת טיב שחזור הנתונים-לדוגמא עבור מקור דיסקרטי  $P_e = \Pr(\hat{D}_n \neq D_n) < 10^{-6}$ ,

עבור מקור רציף  $Distortion = \frac{1}{T} \int_0^T (\hat{x}(t) - x(t))^2 dt$  או  $E[(\hat{x}(t) - x(t))^2]$  באופן כללי

יותר ניתן להכליל למטריקות אחרות מהצורה  $\rho(\hat{x}, x)$  אשר מודדות את העוות בין המקור לאות המשוחזר.

(3) אי ודאות לגבי התנהגות המקור והתווך:

a. פילוג המקור,  $f(x)$  or  $\Pr(D)$ , לא ידוע.

b. פילוג המעבר  $f(r | s)$  אינו ידוע.

דוגמאות:

דוגמאות בתווך אנלוגי-  $r(t) = s(t) * h(t) + n(t)$ . האם יחס אות לרעש ידוע?

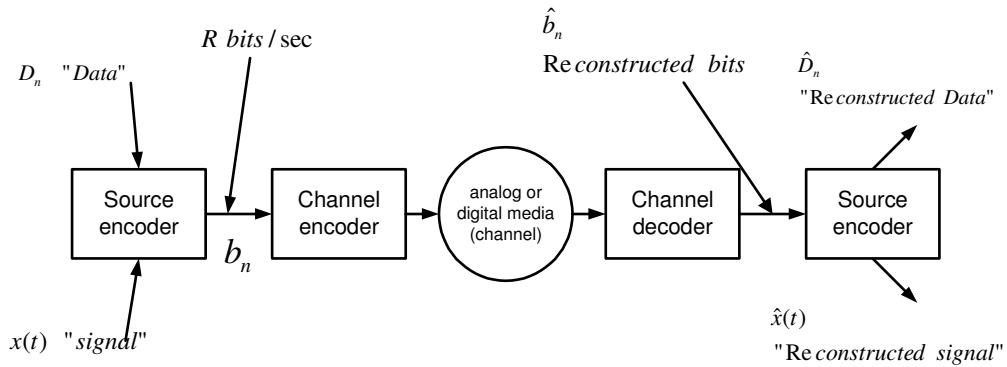
האם פילוג הרעש ידוע?

דוגמאות בתווך ספרתי:

א. מהו הסיכוי לאיבוד חבילות ברשת מחשבים.

ב. מה הסיכוי לשריטה ב CD ומהי תוחלת גודלה.

תרחיש תקשורת ספרתית ועקרון ההפרדה של שאנון:



עקרון ההפרדה:

הערוך מאופיין ע"י  $C \left[ \frac{\text{bits}}{\text{sec}} \right]$ -הקיבול, מספר הביטים אשר ניתן להעביר בערוץ ללא

שגיאה, למשל עבור ערוץ AWGN ברוחב  $W$ ,  $C = W \log \left( 1 + \frac{P}{N_o W} \right)$ .

מקור מאופיין ע"י  $H[\text{bits/sec}]$ -האנטרופיה או (בד"כ אם המקור רציף) ע"י פונקציית קצב עוות  $R(D)$ , לדוגמא מקור גאوسي לבן ברוחב סרט  $W$  ושונות  $\sigma_x^2$  ומדד עוות

$$R(D) = W \log \left( \frac{\sigma_x^2}{D} \right) \text{ bits/sec}$$

תוצאות משפטים מרכזיים מתורת האינפורמציה:

לכל מערכת תקשורת ניתן להעביר נתונים באופן אמיין אם  $H < C$  (ללא עוות) או

$R(D) < C$  (עם עוות), כלומר:

$$H < R < C$$

or

$$R(D) < R < C$$

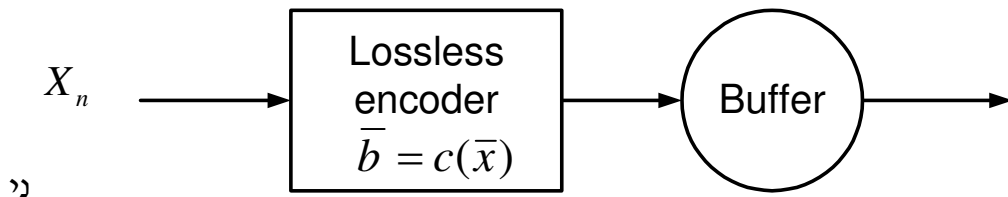
↓

$$D_{\min} = R^{-1}(C)$$

משום שפונקציית קצב העוות  $R(D)$  היא מונוטונית לא-עולה של העוות  $D$ .

## דחיסה ללא עוות

פעולת הדוחס: הפעלת פונקציה קידוד המתאימה לכל וקטור מקור  $\bar{x}$  וקטור  $\bar{b}$  מעל האלפבית של הקוד:  $c: (x_1, x_2, \dots, x_n) \rightarrow (b_1, b_2, \dots, b_k)$  כאשר היחס  $k/n$  איננו בהכרח קבוע. בכדי לעבוד עם ערוץ עם קיבול קבוע בזמן עלינו להוסיף buffer בכדי לפלוט קצב קבוע של ביטים.



תן לסווג למספר סוגי קידוד:

(1) קידוד בקצב קבוע:  $k$  ויח קבועים מראש. במקרה זה נאמר כי הקוד עובד בקצב

$$\text{של } R = \frac{k}{n} [\text{bits / sample}] \text{ (בהנחה שאלפבית הקוד הוא בינארי).}$$

(2) קידוד Block-to-Variable (B2V):  $n$  קבוע מראש,  $k$  תלוי בווקטור  $\bar{x}$  הספציפי:

$$R = \frac{E|c(\bar{x})|}{n} \text{ קצב הקוד במקרה זה הוא } k = |c(\bar{x})|$$

$$(3) \text{ V2B: } k \text{ קבוע מראש, } n \text{ תלוי בכניסה. במקרה זה } R = \frac{k}{E[n]} [\text{bits / sample}]$$

דוגמא למקודד מסוג זה: אלגוריתם זיו-למפל.

(4) V2V.

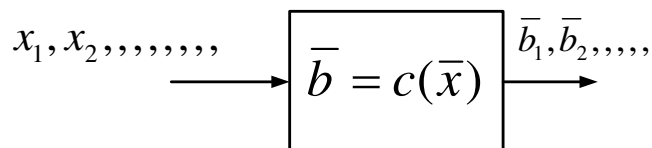
מדוע דוחסים?

- כדאי לדחוס כי בד"כ  $H \ll R_{\text{uncoded}}$  (כלומר ניתן לדחוס).
- חייבים לדחוס כי  $C < R_{\text{uncoded}}$ , כלומר בעזרת המשאבים העומדים לרשותנו לא נצליח להעביר את המקור ללא דחיסה.

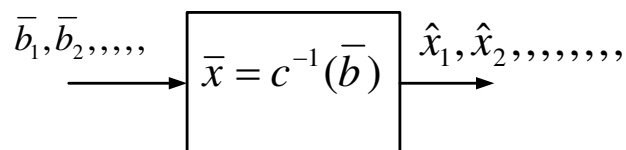
## קידוד בקצב משתנה וחסם האנטרופיה

נניח כי המקור  $X$  מתפלג מעל אלפבית  $\chi$  ו'  $|\chi| = k$ .

פעולת הקידוד:



פעולת המפענח:



ההנחה היא כי אין שגיאה וכי  $\hat{x}_n = x_n \forall n$ . כלומר הפונקציה  $c(x)$  היא חד-חד ערכית

(uniquely decodable).

דוגמא 1:

$\{X\}$  סדרה חסרת זכרון i.i.d. עם פילוג  $(1/4, 1/4, 1/4, 1/4)$ .  $|\chi| = 4$   $\chi = \{0, 1, 2, 3\}$

X	C(X)
0	00
1	01
2	10
3	11

עבור קוד זה, האורך הממוצע הוא:  $R = \bar{L} = \sum p_i l_i = 2$ .

דוגמא 2 :

$$X \sim (0.5, 0.25, 0.125, 0.125)$$

X	C(X)
0	0
1	10
2	110
3	111

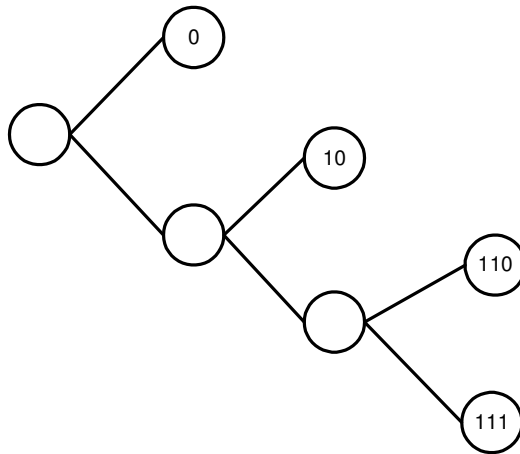
$$R = \bar{L} = \sum p_i l_i = 1.75$$
 עבור קוד זה :

האם ניתן לפענח את הקוד המופיע בדוגמא 2? (כן, זהו קוד prefix כפי שנראה בהמשך).

האם ניתן לדחוס יותר? (לא, תוחלת אורך הקוד היא האנטרופיה).

קודי קידומת (prefix code)

אף מילת קוד היא לא קידומת של מילת קוד אחרת ולכן ניתן לפענח מיד עם סיום מילת קוד. קוד כזה ניתן לתאור ע"י עץ בינארי (עלים הם מילות קוד, צמתים פנימיות אינן).



הערה : לא כל קוד בעל פענוח יחיד הוא בהכרח prefix code.

נגדיר עץ גזום למשעי כעץ אשר מכל צומת פנימית יוצאים זוג ענפים.

### אי-שיויון קראפט:

כוון 1 (הפוך): כל קוד קידומת עם אורכים  $l_1, l_2, \dots, l_k$  מקיים  $\sum_1^k 2^{-l_i} \leq 1$  ושיויון אם"ם העץ גזום למשעי.

כוון 2 (ישר): עבור כל קבוצת אורכים  $l_1, l_2, \dots, l_k$  המקיימת  $\sum_1^k 2^{-l_i} \leq 1$  ניתן למצוא קוד

prefix בינארי שאלו אורכי מילותיו. (הוכחה: ראו Cover).

ניתן להוכיח גם עבור קודים בעלי פענוח יחידי (או חד-חד ערכיים עבור סדרות) וזו בעצם הוכחה כי בחירתנו לעבוד עם קודי קידומת (אשר נבעה מנוחיות פענוח) אינה מגבילה אותנו מבחינת תוחלת אורך קוד לעומת המשפחה הכללית יותר של קודים בעלי פענוח יחיד.

### קידוד שאנון פאנו:

בהנתן הסתברויות  $p_1, \dots, p_k$  נבחר אורכים  $l_i = \lceil -\log p_i \rceil$ .

נבדוק כי בחירה זו מקיימת את אי שיויון קראפט:

$$\sum_1^k 2^{-l_i} = \sum_1^k 2^{-\lceil -\log p_i \rceil} \leq \sum_1^k 2^{-\log p_i} = \sum_1^k p_i = 1$$

בנוסף ניתן לחסום את תוחלת אורך הקוד:

$$H = -\sum p_i \log p_i \leq \bar{L} = \sum p_i l_i = \sum p_i \lceil -\log p_i \rceil \leq -\sum p_i \log p_i + 1 = H + 1$$

טענה:

לכל קוד קידומת (כלומר, שאורכי מילות הקוד מקימות את אי שיויון קראפט) אשר מקודד מקור עם אנטרופיה  $H$  יתקיים:  $\bar{L} \geq H$  ושיויון אם"ם ההסתברויות של אותיות המקור הן חזקות שלמות של גודל האלפבית של הקוד.

הוכחה:

נניח שהקוד מתייחס לעץ בינארי גזום למשעי (אחרת ניתן ל"ייעל" את הקוד) כלומר

$$\sum_1^k 2^{-l_i} = 1. \text{ נגדיר פילוג חדש: } q_i = 2^{-l_i}$$

$$\bar{L} = \sum_1^k p_i l_i = \sum_1^k p_i \log(1/q_i) = \sum_1^k p_i \log(1/p_i) + \sum_1^k p_i \log(p_i/q_i) = H + D(p \parallel q) \geq H$$

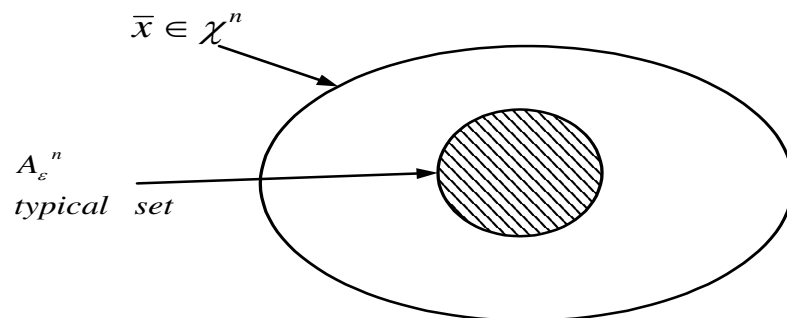
ר  $D(\cdot \parallel \cdot)$  הוא הדיברגנס האינפורמציוני (מרחק kullback-leibler) ושיויון אם"ם

הפילוגים שווים (ראו COVER פרק 2).



מסקנה: קוד קידומת יתאים בדיוק למקור עם פילוג  $p_i = 2^{-l_i}$  ולכן האורך האידיאלי למילת קוד היא האינפורמציה העצמית של הסימבול אותו היא מקודדת,  $I^*(x) = -\log(p(x))$ .

ניתוח אסימפטוטי של שאנון (עבור בלוק אינסופי) והמשפט ההפוך הכללי לקידוד מקור:  
בעזרת AEP:



עבור מקור ארגודי  $x^n$  מוכיחים קיומה של קבוצה אופיינית בעלת התכונות:

$$\Pr(x^n \in A_\epsilon^n) \approx 1 \quad (1)$$

(2) כל הסדרות בקבוצה הן בהסתברות (כמעט) שווה, ושווה ל  $2^{-nH}$ .

מניתוח זה נובע כי קידוד אופטימלי של  $x^n$  נותן אורך קוד זהה לכל סדרה השייכת

לקבוצה האופיינית, ולכן:  $l = \log |A_\epsilon^n| \approx nH \rightarrow \text{rate} = H \text{ bit / symbol}$

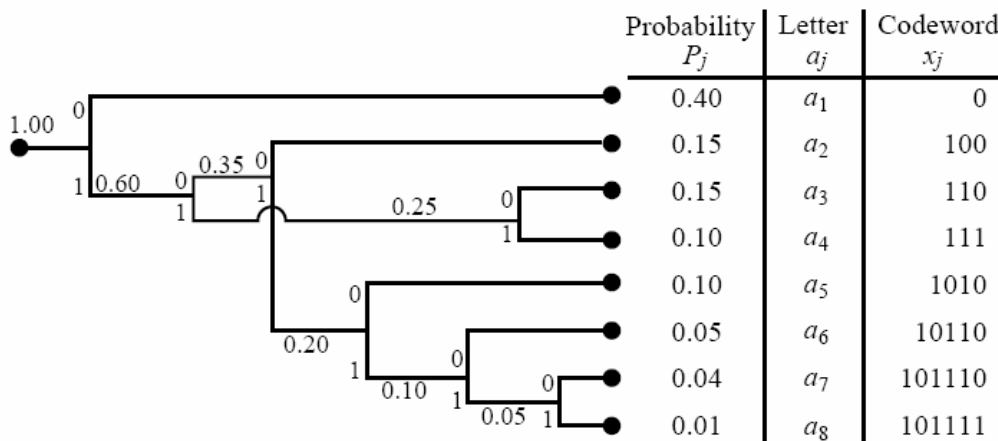
כלומר, עבור סדרות מקור ארוכות שום קוד לא יכול לרדת מקצב האנטרופיה.

קוד האפמן-HUFFMAN:

מה עושים כאשר ההסתברויות אינן חזקות שלמות של 2?

האפמן הראה שיטה איטרטיבית לבניית קוד אופטימלי. בכל שלב נאחד זוג סימבולים

בעלי הסתברות הקטנה ביותר. לאחר  $|\mathcal{X}|-1$  שלבים נגיע לשורש של עץ הקוד.



תכונות הקוד :

1. העץ "גזום למשעי".
2. זוג האותיות הכי נדירות ממופות למילים הכי ארוכות והמילים שונות רק ב-

.LSB

זוג התכונות הנ"ל מתקיים בכל איטרציה במהלך הבניה.

האפמן הראה כי הקוד האופטימלי חייב לקיים :

- (1) עץ גזום למשעי.
- (2) זוג הסימבולים הנדירים ביותר ממופה לזוג המילים הארוכות ביותר וכי ניתן להניח ללא הגבלת כלליות כי מילים אלו שונות רק בLSB.
- (3) בהנתן תכונה 2 מתקיים :

$$\bar{L}(C_{|x|}) = \bar{L}(C_{|x|-1}) + p_{|x|} + p_{|x|-1}$$

כאשר ההנחה היא כי ההסתברויות מסודרות בסדר יורד והקודים השונים הם הקודים המתאימים לאיטרציה האחרונה והלפני אחרונה.

תכונות אלו מוכיחות כי קוד האפמן הוא אופטימלי (למרות שהפתרון אינו בהכרח יחיד). גם עבור קוד האפמן לא ניתן להוכיח חסם הדוק יותר

$$\bar{L}_{Huffman} \leq \bar{L}_{Shannon-Fano} \leq H + 1.$$

כיצד נפתרים מהעודף (יתירות) של 1 ביט? איך משיגים אנטרופיה של מקור עם זכרון?

תשובה: מקודדים ביחד בלוקים ארוכים של  $n$  אותיות מקור ומתכננים את הקוד לפי הפילוג המשותף של  $x_1, x_2, \dots, x_n$ . בדרך זו מקבלים:

$$R \leq \frac{1}{n} [H(X_1, \dots, X_n) + 1] \xrightarrow{n \rightarrow \infty} \text{Entropy rate}$$

מגבלות מעשיות בשימוש בקוד האפמן:

- (1) נדרשת ידיעת הפילוג לצורך תכנון הקוד.
- (2) קוד "טבלאי" ולא אלגוריתמי ולכן צורך זכרון רב.
- (3) בכדי לצמצם את היתירות עלינו לעבוד בבלוקים גדולים וספר הקוד גדל בצורה אקספוננציאלית באורך הבלוק.

דרוש קוד אלגוריתמי!

# Arithmetic coding, Elias Code

Summarized: Ronit Bustin

March 27, 2005

## 1 References

1. Rissanen, J. 1976. Generalized Kraft inequality and arithmetic coding. IBM Journal of Research and Development 20:198-203
2. Pasco, R. 1976. Source coding algorithms for fast data compression. Ph.D. dissertation. Stanford University.
3. Gallager, R.G. unpublished notes.
4. Cover and Thomas, 1991. Elements of Information Theory. Chapter 5.
5. Gibson J.D., Berger T. Lookabaugh T. Lindbergh D. Baker R.L. Digital Compression for Multimedia. 1998. pages 95-102.

## 2 The goal

The goal of arithmetic coding is finding an algorithmic coding scheme (source word  $\rightarrow$  code word), and not a table based coding, without compromising optimality, that is, still reaching the entropy in code length.

## 3 The main idea for coding a source sequence $n \rightarrow \infty$

1. Each source sequence  $x_1, x_2, \dots = X^\infty$  is mapped to a single point  $y$  in the interval  $[0, 1)$ .
2. The point  $y$  can also be described by its binary development:  $y = 0.b_1b_2b_3 \dots$
3. The code word is given by  $b_1b_2b_3 \dots$
4. The goal is to receive an incompressible code-book. This is reached by making use of our knowledge of the source's probabilistic structure. The mapping  $X^\infty \rightarrow y$  should create a uniform distribution of  $y$  in the interval  $[0, 1)$ .  
Reminder:  $y = F_x(x)$  has a uniform distribution in the interval  $[0, 1)$ .
5. The construction of the code is sequential:

- a. A source sequence of length  $n$ , will be mapped to the interval  $J(x^n)$ , so that:

$$\bar{x}_1 \neq \bar{x}_2 \implies J(\bar{x}_1) \cap J(\bar{x}_2) = \emptyset$$

- b. The mapping of a source sequence of length  $n + 1$  will be a refinement of the mapping done for a source sequence of length  $n$ .
- c. For each  $n$ , we require that a “more likely” source sequence of length  $n$  will be mapped to a larger section in the interval  $[0, 1)$ , so that it could be described in less bits.
- d.  $J(x_n) \xrightarrow{n \rightarrow \infty} y$

## 4 Definitions

1. A binary development:

$$0.b_1b_2b_3 \dots = \sum_{n=1}^{\infty} b_n \cdot \frac{1}{2^n} \quad \forall b_i \in \{0, 1\}$$

2. A development over a general alphabet  $\chi$ :

$$0.x_1x_2x_3 \dots = \sum_{n=1}^{\infty} x_n \cdot \frac{1}{|\chi|^n} = \text{value}(X^\infty) \quad \forall x_i \in \chi$$

3. Partitioning of the interval  $[0, 1)$  to binary sections:

$$J_B(b^n) = [0.b_1b_2 \dots b_n, 0.b_1b_2 \dots b_n + 2^{-n}) = \text{half open section of size } 2^{-n}$$

4. Partitioning of the interval  $[0, 1)$  according to the source  $\chi$  (with a known distribution  $p(x^n)$ ):

$$J(x^n) = [ f(x^n), f(x^n) + p(x^n) ) \quad \text{where, } f(x^n) = \Pr\{\text{value}(X^n) < \text{value}(x^n)\}$$

## 5 Example - Phase one

We will examine a memoryless trinary source:

$$\chi = \{0, 1, 2\}$$

$$p(x) = \begin{cases} x = 0, & 0.6 \\ x = 1, & 0.2 \\ x = 2, & 0.2 \end{cases}$$

Figure 1, demonstrates the sections received for the first two source letters, and for the first two code bits.

$J_0^{(00)}$	$J_0^{(01)}$	$J_0^{(10)}$	$J_0^{(11)}$	Code sequence: n = 2					
$J_0^{(0)}$		$J_0^{(1)}$		Code sequence: n = 1					
$J^{(0)}$			$J^{(1)}$	$J^{(2)}$	Source sequence: n = 1				
$J^{(00)}$	$J^{(01)}$	$J^{(02)}$	$J^{(10)}$	$J^{(11)}$	$J^{(12)}$	$J^{(20)}$	$J^{(21)}$	$J^{(22)}$	Source sequence: n = 2

Figure 1: Partitioning of  $[0,1)$  for both the first two source letters and the first two code bits

## 6 Specific case - memoryless source with alphabet $\chi$

For this case we have a recursive formula for the partitioning according to the source:

$$\text{a memoryless source: } p(x^n) = \prod_{i=1}^n p(x_i)$$

$$f(x^{n+1}) = f(x^n) + f_1(x_{n+1}) \cdot p(x^n) \quad \text{where, } f_1(x) = \begin{cases} \sum_{i=0}^{x-1} p(i), & x \geq 0 \\ 0, & x = 0 \end{cases}$$

## 7 Remarks & conclusions

1.  $|J(x^n)| = p(x^n)$
2.  $\sum_{x^n} J(x^n) = 1$
3. The lower bound =  $f(x^n) \leq f(x^{n+k})$  for all  $k$   
The upper bound =  $f(x^n) + p(x^n) > f(x^{n+k})$  for all  $k$
4. Nesting of intervals:

$$\forall n \quad J(x^{n+1}) \subseteq J(x^n) \iff x^{n+1} = (x^n, x_{n+1})$$

That is,

$$J(x_1) \supseteq J(x_1x_2) \supseteq J(x_1x_2x_3) \supseteq \dots \supseteq J(X^\infty) = f(X^\infty)$$

5.  $y = f(X^\infty) \sim \text{Unif}(0, 1)$

## 8 Sequential Coding & Decoding of an infinite length source sequence

The main principle is to code & decode “on the fly”

1. At any given time, the coder will put out only the bits (from the partial binary development) that he can commit on, that is:  $J(x^n) \subseteq J_B(b^l) \Rightarrow b^l$
2. At any given time, the decoder will reconstruct only the source letters that he can commit on, that is:  $J_B(b^l) \subseteq J(\hat{x}^m) \Rightarrow \hat{x}^m$

The system will have a delay both in coding, and decoding.

## 9 Coding & Decoding rates

- The coding rate is:

$$\begin{aligned} J(x^n) &\subseteq J_B(b^l) \\ &\downarrow \\ p(x^n) = |J(x^n)| &\leq |J_B(b^l)| = 2^{-l} \\ &\downarrow \\ l &\leq -\log(p(x^n)) = I(x^n) \end{aligned}$$

- The decoding rate is:

$$\begin{aligned}
 J(x^m) &\supseteq J_B(b^l) \\
 &\downarrow \\
 p(x^m) = |J(x^m)| &\geq |J_B(b^l)| = 2^{-l} \\
 &\downarrow \\
 l &\geq -\log(p(x^m)) = I(x^m)
 \end{aligned}$$

- To conclude: At time  $n$  we code below the *self-information* of the source up to that time (since we have not coded all of  $x^n$ ), but we decode above the *self-information* of the output up to that time.
- The delay:
  - a. The delay of *information* in the coder is:  $I(x^n) - l$
  - b. The delay of *information* in the decoder is:  $l - I(\hat{x}^m)$
  - c. The total delay in source letters is:  $n - m$

*Gallager's Theorem:* For every memoryless source the delay is:

$$E\{I(x^n) - I(\hat{x}^m)\} \leq \text{Bound}(p_1, p_2, \dots, p_{|\chi|})$$

## 10 Example - Phase two

We will look again at the memoryless trinary source:

$$\begin{aligned}
 \chi &= \{0, 1, 2\} \\
 p(x) &= \begin{cases} x = 0, & 0.6 \\ x = 1, & 0.2 \\ x = 2, & 0.2 \end{cases}
 \end{aligned}$$

$J_{\frac{1}{2}}(00)$	$J_{\frac{1}{2}}(01)$	$J_{\frac{1}{2}}(10)$			$J_{\frac{1}{2}}(11)$			Code sequence: n = 2	
$J_{\frac{1}{2}}(0)$		$J_{\frac{1}{2}}(0)$						Code sequence: n = 1	
$J(0)$			$J(1)$			$J(2)$			Source sequence: n = 1
$J(00)$	$J(01)$	$J(02)$	$J(10)$	$J(11)$	$J(12)$	$J(20)$	$J(21)$	$J(22)$	Source sequence: n = 2

Figure 2: Partitioning of  $[0,1)$  for both the first two source letters and the first two code bits

For this source we will now detail the coder and decoder steps, when sending a source sequence "010...":

For the coder:

1. “0” -

$$y \in J(0) = [0, 0.6) \implies y \in (J_B(0) \vee J_B(1))$$

We can not decide on coding bits.

2. “01” -

$$y \in J(01) = [0.36, 0.48) \implies y \in J(01) \subset J_B(01) = [0.25, 0.5)$$

We can decide on coding bits - “01”

3. “010” -

$$y \in J(010) = [0.36, 0.432) \implies y \in (J_B(010) \vee J_B(011))$$

We have no additional information - no additional coding bits to send.

For the coder:

1. “01” - received from the coder

$$y \in J_B(0) = [0.25, 0.5) \implies y \in J(0) \subset J_B(0)$$

We can decide only on the first source letter = “0”. But:

$$y \in (J(00) \vee J(01))$$

We can not decide on any more source letters.

## 11 Practical aspects

1. A source with memory: the mechanism is the same. The only difference is that the partitioning is based on the past, that is,  $Pr(x_n|x_{n-1}, \dots, x_1)$
2. Implementation aspects:

- The coding is precision-hungry: requires growing levels of precision in order to determine relativity to an interval.
- The coding is retentive: code symbols sometimes are hoarded and released highly asynchronously in variable-length bursts.

These two shortcomings of the arithmetic coding are tightly intertwined. Solving the coder’s tendency to be retentive will allow *rescaling*, that will solve it’s precision-hungriness.

Solutions:

- Solution for retentivity: coder & decoder will agree upon a protocol for such cases, i.e., setting a limit on the maximum precision that will be permitted, whenever that limit is about to be exceeded they will stuff into the data stream a fictional data symbol that will end retention.
- *Rescaling* - Solution for precision-hungriness: after the source commits & transmits code bits, it then focuses on a specific sub-interval of  $[0, 1)$ . Instead, we can stretch the chosen interval back to  $[0, 1)$ . Example:  
consider a Bernoulli( $\frac{3}{4}$ ), and a source sequence: 1100 ...  
first letter: 1  $\longrightarrow$   $[\frac{1}{4}, 1)$  - no output bits



second letter:1  $\rightarrow [\frac{7}{16}, 1)$  - no output bits

third letter:0  $\rightarrow [\frac{7}{16}, \frac{37}{64}) = [0.4375, 0.578125)$  - no output bits

fourth letter:0  $\rightarrow [\frac{7}{16}, \frac{121}{256}) = [0.4375, 0.47265625)$  - and finally the coder can emit three code digits 011

Now lets *rescale*:

first code digit:0  $\rightarrow$  the coder is operating in an interval contained entirely in the left half of  $[0, 1)$ . This will allow both coder and decoder to eliminate  $[0.5, 1)$  and scale everything in the left half by a factor of two:

$$[\frac{7}{16}, \frac{121}{256}) = [0.4375, 0.47265625) \Rightarrow [\frac{7}{8}, \frac{121}{128}) = [0.875, 0.9453125)$$

second code digit:1  $\rightarrow$  the coder is operating in an interval contained entirely in the right half of what has been rescaled to fill  $[0, 1)$ . This will allow both coder and decoder to eliminate  $[0, 0.5)$  and scale everything in the right half by a factor of two. It is more convenient to shift  $[0.5, 1)$  down to  $[0, 0.5)$  before upscaling by two:

$$[\frac{7}{8}, \frac{121}{128}) = [0.875, 0.9453125) \Rightarrow [2 \cdot (\frac{7}{8} - 0.5), 2 \cdot (\frac{121}{128} - 0.5)) = [\frac{3}{4}, \frac{57}{64}) = [0.75, 0.890625)$$

second code digit:1  $\rightarrow$  again we will shift and upscale by two:

$$[\frac{3}{4}, \frac{57}{64}) = [0.75, 0.890625) \Rightarrow [2 \cdot (\frac{3}{2} - 0.5), 2 \cdot (\frac{57}{64} - 0.5)) = [\frac{1}{2}, \frac{25}{32}) = [0.5, 0.78125)$$

3. The coding is parametric: requires the knowledge of the source's distribution. The solution is empiric distribution:

For a memoryless source:  $\hat{P}^{(n)}(x) = \text{func}\{x_1, x_2, \dots, x_n\}$

For a Markov source:  $\hat{P}^{(n)}(x_{i+1}|x_i) = \text{func}\{x_1, x_2, \dots, x_n\}$

4. Coding a source sequence of finite length:  $C(x^n)$

- Case one: We want to allow concatenation of blocks. This requires the fulfilment of the prefix free condition, that is,  $J_B(b^l) \subseteq J(x^n)$ . This is accomplished by:

$$C(x) = \lfloor \bar{F}(x) \rfloor_{l(x)}$$

$$\text{where, } l(x) = \lceil -\log(p(x)) \rceil + 1$$

$$\text{where, } \bar{F}(x) = F(x) - \frac{p(x)}{2} = f(x) + \frac{p(x)}{2}$$

This will give us an addition of two bits over the *entropy*.

See Cover & Thomas [4].

- Case two: "one shot" coding, coding of a single block. No fulfilment of the prefix free condition needed.  $0.b^l \in J(x^n)$

$$C(x) = \lfloor F(x) \rfloor_{l(x)}$$

$$\text{where, } l(x) = \lceil -\log(p(x)) \rceil$$

This will give us an addition of one bits over the *entropy*.

See Gibson et al., [5].

## UNIVERSAL LOSSLESS SOURCE CODING

Lecture notes for the course *Data Compression*, faculty of engineering, TAU.  
Summary by *Amir Ingber*.

### 1. PROBLEM DESCRIPTION

How do we code a sequence  $x_1, x_2, \dots, x_n$ , when the distribution  $p(x_1^n)$  is unknown (or no-existent)?

When the data  $x_1, x_2, \dots, x_n$  comes from an unknown stochastic source  $p_\theta(x)$ ,  $\theta \in \Theta$ , the setting is called the *stochastic* or *probabilistic setting*. when there is no stochastic source for the data, i.e. the data is a single, individual sequence of symbols, we call it the *deterministic setting*.

This summary deals with the probabilistic setting.

### 2. PARAMETRIC FAMILIES OF SOURCES

It is usually convenient to assume a parametrization on the sources:  $p \in \{p_\theta : \theta \in \Theta\}$ . Here knowing the source  $p$  is equivalent to knowing  $\theta$ .

Examples of parametric families of sources:

**Binary memoryless sources:**

$$X \sim \text{Ber}(\theta), 0 \leq \theta \leq 1.$$

**Markov chain with finite alphabet:**

Here  $\underline{\theta}$  is the transition probability matrix of the chain,  $p_{ij}$ , where  $0 \leq i, j \leq |\mathcal{X}| - 1$ . Usually stationary sources are assumed, therefore the initial state of the chain is neglected.

**Autoregressive sources with finite resolution:**

The source  $X$  is generated according to the following model:

$$X_n = \sum_{i=1}^q \theta_i X_{n-i} + W_n$$

where  $W_n$  is the generating white noise, and the vector  $\underline{\theta}$  consists of the IIR filter coefficients.

The optimal rate for decoding a source is its *entropy*:

$$\text{Rate} = \frac{1}{n} H_\theta(X^n) = \frac{1}{n} \sum_{x^n} p_\theta(x^n) \log \frac{1}{p_\theta(x^n)}$$

where  $p_\theta(\cdot)$  is the source's real probability function.

### 3. TWO APPROACHES IN UNIVERSAL CODING

These are two possible models for a universal coding system:

**3.1. The batch approach.** In the batch approach the processing is done per block. First, the parameters of the process are estimated from the block, and then coded assuming these parameters.

Since the decoder knows neither the estimated parameters nor the data that the parameters are estimated from, the decoder also adds the estimated parameters to the codeword as *side-information* (See Fig. 1(a)).

The effective code length (in bits per source symbol) is

$$\frac{1}{n} [\ell(\text{side information}) + \ell(\text{code}(x_1^n|\theta))].$$

Due to the nature of working in block units, a delay of one block is inevitable. The batch approach is also called the *2-pass* or *2-part* approach, for obvious reasons.

**3.2. The sequential approach.** Also called the *adaptive* approach. Here the parameters for coding a symbol are estimated based on *past* data (See Fig. 1(b)).

Since the code is lossless, the decoder can estimate the process parameters in the same method used in the encoder, and by that eliminating the need for the side information as in the batch approach.

On the face of it, the sequential approach seems preferable since side information is not needed. However, it should be kept in mind that the parameters used for coding each symbol are estimated based on past data and not on the symbol itself, resulting in a longer code length.

Having estimated the parameters  $\hat{\theta} = \hat{\theta}(x_1^{i-1})$ , the probability used for coding the symbol  $x_i$  is  $\hat{p}(x|x_1^{i-1}) = p_{\hat{\theta}}(x|x_1^{i-1})$ , and the ideal code length for one symbol is

$$\ell_u(x_i|x_1^{i-1}) = -\log \hat{p}(x_i|x_1^{i-1}).$$

This can be achieved, for example, by an adaptive arithmetic code.

The code length for the entire sequence is given by:

$$\ell_u(x_1^n) = \sum_{i=1}^n \ell_u(x_i|x_1^{i-1}) = -\log \prod_{i=1}^n \hat{p}(x_i|x_1^{i-1}) \triangleq -\log p_u(x_1^n).$$

In fact, the code length is induced by some probability function  $p_u(\cdot)$  over  $\mathcal{X}^n$ , which is called the *universal* distribution.

As opposed to the batch approach, where the convergence to the distribution of the source is done in advance, in the sequential approach the universal probability  $p_u(\cdot)$  converges to the true source's distribution on-the-fly, as the sequence progresses.

#### 4. REDUNDANCY IN UNIVERSAL CODING

Given a universal decoder  $u$ , that maps source words  $x_1^n$  to code words of length  $\ell_u(x_1^n)$ , its redundancy over the entropy of the source is given by:

$$R_n(\theta, \ell_u) \triangleq \frac{1}{n} \left[ \underbrace{\sum_{x_1^n} p_{\theta}(x_1^n) \ell_u(x_1^n)}_{\text{Average universal codelength}} - \underbrace{H_{\theta}(X^n)}_{\text{Entropy bound}} \right]$$

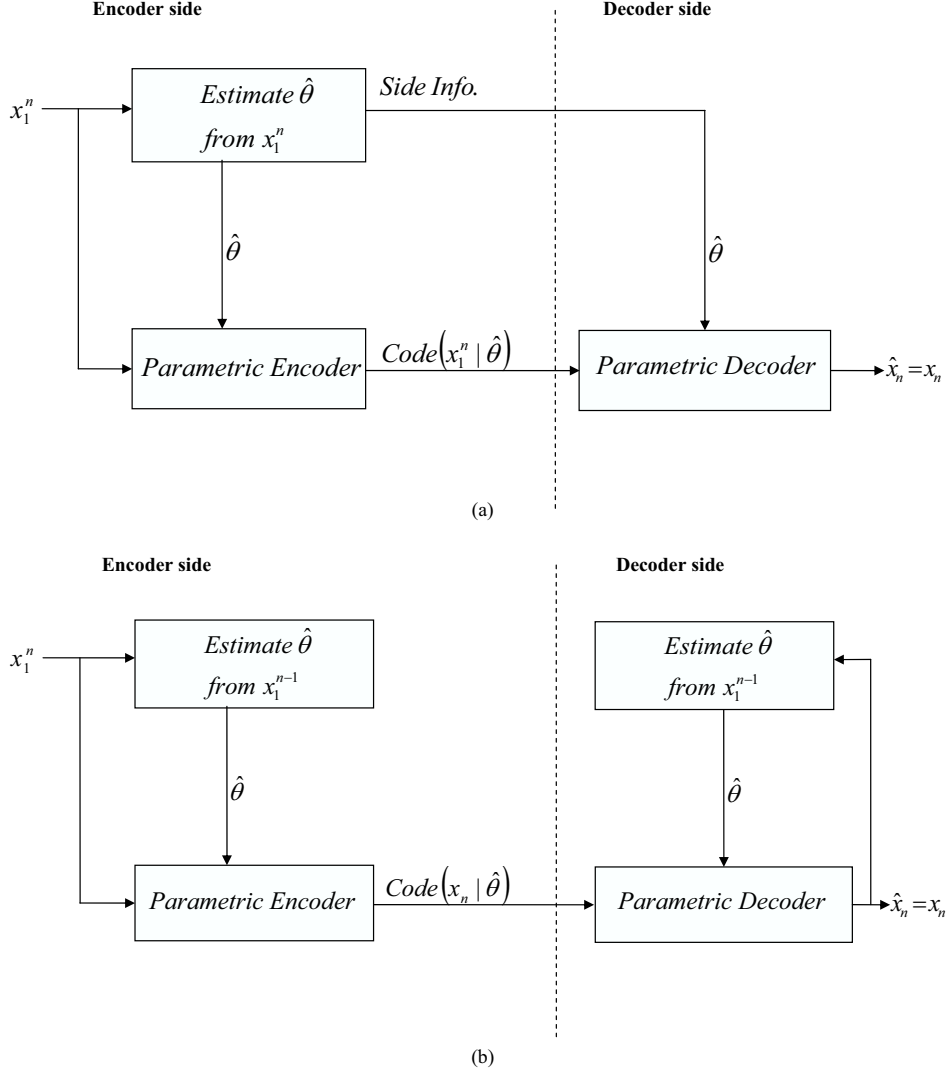


FIGURE 1. Two approaches in universal coding: (a) The batch approach. (b) The sequential approach

A coder is called *universal* w.r.t. a group of sources  $\Theta$  if

$$\forall_{\theta \in \Theta} \lim_{n \rightarrow \infty} R_n(\theta, \ell_u) = 0.$$

If the convergence is uniform in  $\Theta$ , then the encoder is called *strongly universal*. If not, it is called *weakly universal*

## 5. FEASIBILITY OF UNIVERSAL CODING: AEP VIEWPOINT

As we know, it is possible to code a source  $X$  in a rate  $R = H(X)$  when the source is known - e.g. Huffman coding. Now, suppose we wish to code with a constant

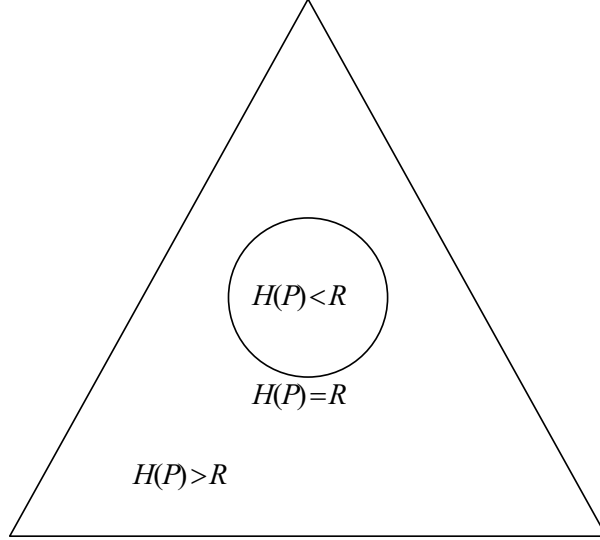


FIGURE 2. The probability simplex

rate  $R$ . We now show, that it is possible to universally code all memoryless sources with entropy  $H \leq R$ . This is based on the fact that the number of sequences of a type  $P$  increases exponentially, and that there is only a polynomial number of types.

We consider all the memoryless sources with alphabet  $\mathcal{X}$ , that is the *simplex*  $\{p_1, p_2, \dots, p_{|\mathcal{X}|} | p_i \geq 0, \sum_i p_i = 1\}$ .

Inside that simplex, we see the sources with entropy equal to, greater and less than  $R$ . (See Fig (2)).

define the set  $A$  as the set of all sequences, whose type's entropy is less than or equal to  $R$ :

$$A \triangleq \{\mathbf{x} \in \mathcal{X}^n | H(P_{\mathbf{x}}) \leq R\}.$$

Then, by marking the set of all types by  $\mathcal{P}$ :

$$|A| = \sum_{P \in \mathcal{P}: H(P) \leq R} |\mathcal{T}(P)| \leq \sum_{P \in \mathcal{P}: H(P) \leq R} 2^{nH(P)} \leq \sum_{P \in \mathcal{P}: H(P) \leq R} 2^{nR}$$

By noting that the number of possible types of sequences from  $\mathcal{X}^n$  is  $(n+1)^{|\mathcal{X}|-1}$ , we get :

$$|A| \leq (n+1)^{|\mathcal{X}|-1} 2^{nR}$$

The needed rate to encode all the sources from  $A$  is :

$$\frac{1}{n} \log_2 \bigcup_{P \in A} |\mathcal{T}(P)| = \frac{1}{n} \log_2 (n+1)^{|\mathcal{X}|-1} 2^{nR} = \frac{|\mathcal{X}|-1}{n} \log_2 (n+1) + R$$

Which goes to  $R$  as  $n \rightarrow \infty$ , meaning that coding of all sources of entropy  $H(X) < R$  can be coded at rate  $R$ .

The section is based also on [1], the result appears also in [2].

## 6. AN EXAMPLE: VARIABLE RATE UNIVERSAL CODING OF A BERNOULLI SOURCE

The objective is to code a binary sequence  $x_1^n = x_1, x_2, \dots, x_n$  when the distribution is unknown. It is only known that the sequence comes from a Bernoulli source, with unknown  $p$ .

### 6.1. The batch approach.

First, we count the number of ones in the sequence, marked by  $k$ . We send  $k$  as side-information to the decoder. Since  $k$  is taken from  $\{0..n\}$ , this adds to the overall rate of the code a factor of  $\lceil \log_2(n+1) \rceil$  bits. We now select one of two options (which lead to the same asymptotic rate):

- Assume that the source is Bernoulli( $\frac{k}{n}$ ) and use Huffman or arithmetic code to encode the sequence.

By assuming that the sequence is long enough,  $\frac{k}{n}$  converges to the real  $p$  of the sequence, and the rate of the code converges to  $h(p)$  bits per source symbols, plus an additive term that vanishes like  $O(\frac{1}{n})$ . To that rate we must add the rate of the side information -  $\lceil \log_2(n+1) \rceil$  for the entire sequence, or  $\frac{1}{n} \lceil \log_2(n+1) \rceil$  bits per source symbol. The overall rate we get is :

$$Rate = h(p) + O\left(\frac{1}{n}\right) + \frac{1}{n} \lceil \log_2(n+1) \rceil \approx h(p) + \frac{1}{n} \log_2(n)$$

The redundancy is given by  $R = Rate - H(X) = \frac{1}{n} \log_2(n)$ .

In fact (shown in [6]), when coding according to Bernoulli( $\theta$ ), it is sufficient to describe  $\theta$  with a resolution of  $\frac{1}{\sqrt{n}}$ , and by that achieving a redundancy of  $\frac{1}{2n} \log_2(n)$ .

- Since the decoder already knows that the sequence has  $k$  ones, it only needs to know which of the  $\binom{n}{k}$  sequences with  $k$  ones the encoder wishes to encode. This requires  $\lceil \log_2 \binom{n}{k} \rceil$  bits for the entire sequence. By utilizing Stirling's approximation:

$$\log_2 \binom{n}{k} \approx n \cdot h(p) - \frac{1}{2} \log_2(2\pi np(1-p))$$

$$\frac{1}{n} \log_2 \binom{n}{k} \approx h(p) - \frac{1}{2n} \log_2(2\pi np(1-p)) \approx h(p) - \frac{1}{2n} \log_2(n)$$

we get the overall code rate:

$$Rate \approx h(p) - \frac{1}{2n} \log_2(n) + \frac{1}{n} \log_2(n) = h(p) + \frac{1}{2n} \log_2(n)$$

(where the term  $\frac{1}{n} \log_2(n)$  is the side information rate). The redundancy is given by  $\frac{1}{2n} \log_2(n)$ .

### 6.2. The sequential approach.

In that approach the estimation for  $p$  is done on the fly, as the sequence progresses.

We mark the *empirical probability after seeing  $i$  bits* as  $\hat{p}(x|x_1^i) \triangleq \hat{p}_i(x)$ . Determining the code length for the next bit is done according to:

$$\ell_u(x_i|x_1^{i-1}) \triangleq -\log_2(\hat{p}_{i-1}(x_i))$$

But how do we determine  $\hat{p}_i(x)$ ?

A somewhat intuitive suggestion might be

$$\hat{p}_i(x) = \begin{cases} \frac{n_0(x_1^i)}{i}, & x=0; \\ \frac{n_1(x_1^i)}{i}, & x=1; \end{cases}$$

where  $n_b(x_1^i)$  marks the number of  $b$ 's ( $b \in \{0, 1\}$ ) in  $x_1^i$ . The motivation for such estimation may come from the fact that as the sequence gets longer,  $\frac{n_b(x_1^i)}{i}$  converges to the real  $p$ .

However, when the coding starts, serious problems occur which are demonstrated by the following simple example:

suppose  $x_1^n = 010\dots$ . First, in the very beginning of the sequence,  $\hat{p}_0(x)$  is undefined. Let us then define  $\hat{p}_0(x) \triangleq \frac{1}{2}$ . The code length for the first symbol shall be  $-\log_2(1/2) = 1\text{bit}$ . For the next symbol, however,  $\hat{p}_1(1) = 0$ , since  $n_1('0') = 0$ , yielding  $\ell_u(x_2|x_1) = \infty$  which is, of course, unacceptable.

We would like an estimator, that starts with  $\hat{p}_0(x) = \frac{1}{2}$ , and continues towards the real probability  $p$  in a more 'relaxed' manner.

A better suggestion is the Laplace estimation for the probability:

$$\hat{p}_i(x) = \begin{cases} \frac{n_0(x_1^i)+1}{i+2}, & x=0; \\ \frac{n_1(x_1^i)+1}{i+2}, & x=1; \end{cases}$$

Laplace's estimator can be thought of as the original estimator, where the string '01' is added at the beginning of the sequence, as a starting condition:

$$\hat{p}_i(x) = \frac{n_x(01x_1^i)}{i+2}.$$

For example:

$x_i$	0	1	0	0	1	0	0	...
$\hat{p}_{Laplace}$	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{2}{4}$	$\frac{3}{5}$	$\frac{2}{6}$	$\frac{4}{7}$	$\frac{5}{8}$	...

and the *universal probability* assigned to the above sequence, defined

$$p_u(x_1^n) \triangleq \prod_{i=1}^n \hat{p}(x_i),$$

is  $\frac{1 \cdot 1 \cdot 2 \cdot 3 \cdot 2 \cdot 4 \cdot 5}{2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 \cdot 8} = \frac{120}{40320}$ .

The code length assigned for a sequence is:

$$\ell_u(x_1^n) = \sum_{i=1}^n \ell_u(x_i|x_1^{i-1}) = -\log_2\left(\prod_{i=1}^n \hat{p}(x_i)\right) = -\log_2(p_u(x_1^n)).$$

For Laplace's estimator, it can be shown that

$$p_u(x_1^n) = \frac{n_1(x_1^n)! \cdot n_0(x_1^n)!}{(n+1)!}$$

which is followed by

$$\ell_u(x_1^n) = -\log_2(p_u(x_1^n)) = \log_2(n+1) + \log_2\left(\binom{n}{n_1(x_1^n)}\right)$$

yielding the same rate and redundancy as the second batch coding method from 6.1 above.

Laplace's estimator is not the only estimator that starts with  $\hat{p}_0(x) = \frac{1}{2}$  and continues in a consistent way; all estimators of the form  $\hat{p}_i(x) = \frac{n_x(x_1^i) + \beta}{i + 2\beta}$ , ( $\beta > 0$ ) share that property. See, e.g. Krichevsky and Trofimov's [4] estimator -

$$\hat{p}_i^{KT}(x) = \frac{n_x(x_1^i) + \frac{1}{2}}{i + 1}.$$

### 6.3. Conclusions and generalizations.

- (1) The additive term  $\frac{\log_2(n)}{2n}$ , which stems from the side information in the batch approach, and from the bias in the sequential approach, does not depend on the sequence.
- (2) The universal distribution  $p_u(x_1^n)$  can be thought of as a mixture over types:

$$p_u(x_1^n) = \frac{1}{n+1} \sum_{\theta \in \{0, \frac{1}{n}, \frac{2}{n}, \dots, \frac{n}{n}\}} \text{Uniform}(\mathcal{T}(\theta))$$

Where  $\mathcal{T}(\theta)$  is the *type*  $\theta$ , i.e. the set of all sequences  $x_1^n$  whose empirical probability is  $\theta$ ; and  $\text{Uniform}(\mathcal{T}(\theta))$  is a probability function over  $\mathcal{X}^n$  that gives an equal probability to sequences from  $\mathcal{T}(\theta)$  and zero probability to other sequences.

This gives:

$$p_u(x_1^n) = \frac{1}{n+1} \text{Uniform}\left(\mathcal{T}\left(\frac{n_1(x_1^n)}{n}\right)\right).$$

From the AEP property, as  $n \rightarrow \infty$ , 'all' of the sequences coming from a source  $\text{Ber}(p)$  are of the type  $\theta \approx p$ , giving :

$$p_u(x_1^n) \approx \frac{1}{n+1} \sum_{\theta \in \{0, \frac{1}{n}, \dots, \frac{n}{n}\}} \text{Ber}(\theta).$$

- (3) A universal distribution may be a mixture of the distributions in the class  $\Theta$ . Note that a mixture of memoryless sources may be with memory.
- (4) Generalizing: when  $\theta$  gets a value out of  $M$  possible values, i.e.  $|\Theta| = M$ , the maximum likelihood (ML) estimator for  $\theta$  is given by:

$$\hat{\theta}_{ML}(x_1^n) = \arg \max_{\theta_i} (p_{\theta_i}(x_1^n)).$$

The coding scheme: First, we encode  $\theta_i$  - which requires  $\lceil \log_2(M) \rceil$  bits. Then we encode  $x_1^n$  assuming that  $\theta = \hat{\theta}$ , using code lengths of  $\lceil -\log_2(p_{\hat{\theta}}(x_1^n)) \rceil$ . Note that for every sequence  $x_1^n$  that comes from a source  $\theta$ :

$$-\log_2(p_{\hat{\theta}}(x_1^n)) \leq -\log_2(p_{\theta}(x_1^n)) = I(x_1^n)$$

which gives

$$E_{\theta}[\ell_u(x_1^n)] \leq \log_2(M) + H_{\theta}(x_1^n)$$



yielding a redundancy  $R_n(\theta, \ell_u) \leq \frac{\log_2(M)}{n}$ . (Remember that  $M$  is dependant on  $n$ .)

(5) The equivalent universal distribution (for 4) :

$$p_u(x_1^n) = \frac{1}{M} \sum_{\theta_i} p_{\theta_i}(x_1^n) \approx \frac{1}{M} \max_{\theta_i} p_{\theta_i}(x_1^n) = \frac{1}{M} p_{\hat{\theta}}(x_1^n)$$

The  $\frac{1}{M}$  factor causes the universality price.

#### 6.4. Further questions.

- What is the optimal resolution for describing  $\theta$ , i.e. how large must  $M$  be?
- Should we always encode  $\hat{\theta}$  according to a uniform distribution (using  $\log_2(M)$  bits)? Can we save a few bits here?
- We know that universal coding is possible if and only if the sources family  $\Theta$  is not too rich, that is if the redundancy vanishes,  $\frac{1}{n} \log(M_n) \rightarrow 0$ . What conditions on the family  $\Theta$  enable this?

### 7. THE LYNCH - DAVISSON THEORY

This approach [3] was continued and broadened by Rissanen [6], Gallager, Feder and Merhav [5] and others.

We deal with a variable rate code, for minimum redundancy.

Suppose we have a B2V code  $\ell(x_1^n)$ , satisfying Kraft's inequality with an equality. (i.e. there exists a distribution  $q(x)$  s.t.  $\ell(x_1^n) = -\log_2(q(x_1^n))$  are integers).

The redundancy for this code (for the source  $\theta$ ) is given by

$$R_n(\theta, \ell_u(\cdot)) = \frac{1}{n} [E_\theta[\ell(X^n)] - H_\theta(X^n)] = \frac{1}{n} D(p_\theta || q).$$

How do we find a good distribution  $q$  for the universal problem? that is, a *universal*  $q$  that is good for the entire family of sources  $p_\theta, \theta \in \Theta$ .

**The min-max sense::** we seek the distribution  $q$  that is optimal for the *worst* source  $\theta$ , that is:

$$R_{Min-Max} \triangleq \frac{1}{n} \min_q \max_\theta D(p_\theta || q)$$

**The Bayesian sense::** we assume a prior distribution over the parameters,  $w(\theta)$ , and get:

$$\bar{R}(q, w) = \frac{1}{n} \sum_\theta w(\theta) D(p_\theta || q)$$

The optimal  $q$  for a given  $w(\theta)$ :

$$q_w^* = \arg \min_q \bar{R}(q, w)$$

Now we take the worst distribution  $w^*(\theta)$ , and get the Max-Min Redundancy:

$$R_{Max-Min} = \frac{1}{n} \max_{w(\theta)} \min_q \sum_\theta w(\theta) D(p_\theta || q)$$

Note that the Min-Max redundancy can be also written as

$$R_{Min-Max} = \frac{1}{n} \min_q \max_{w(\theta)} \sum_{\theta} w(\theta) D(p_{\theta} || q) :$$

Define  $\Theta^* \subseteq \Theta$  as the set of all parameters that achieve the worst redundancy in the min-max definition. This maximum is also achieved by any distribution  $w(\theta)$  over  $\Theta^*$ , with  $w(\theta) = 0$  for  $\theta \notin \Theta^*$ .

Under certain conditions, the order (max-min or min-max) does not change the redundancy, i.e. the *duality gap* is zero, and the optimum is achieved in a saddle point. We shall now see that this is the case here.

By noting  $q_w(x) = \sum_{\theta} w(\theta) p_{\theta}(x_1^n)$ , we get:

$$\begin{aligned} F(w, q) &\triangleq \sum_{\theta} w(\theta) D(p_{\theta} || q) \geq \sum_{\theta} w(\theta) D\left(p_{\theta} || \sum_{\theta} w(\theta) p_{\theta}\right) = \sum_{\theta} w(\theta) D(p_{\theta} || q_w) \\ \bar{L}^{Opt} &= \sum_{\theta} w(\theta) H_{\theta}(X) = H(X|\theta) \end{aligned}$$

$$\bar{L}_u = \sum_{\theta} w(\theta) \sum_{x_1^n} p_{\theta}(x_1^n) \underbrace{\ell_u(x_1^n)}_{-\log_2 q(x_1^n)} = \sum_{x_1^n} q_w(x_1^n) \cdot -\log_2(q(x_1^n)) \geq H_{q_w}(X_1^n)$$

with an equality if and only if  $\forall_x q(x) = q_w(x)$ .

$$\bar{L}_u - \bar{L}^{Opt} \geq H(X_1^n) - H(X_1^n|\theta) = I(X_1^n|\theta)$$

- (1) Without knowing  $\theta$  we encode  $x_1^n$  according to the mixture distribution  $q_w(x)$  (that minimizes the average code length).
- (2) for  $q(x) = q_w(x)$ , the redundancy is  $I(X_1^n|\theta)$  - the mutual information between the parameter(s) and the source  $x_1^n$ .

$$\min_q (H(X_1^n) - H(X_1^n|\theta)) = I(X_1^n|\theta)$$

- (3) What is the worst prior  $w^*(\theta)$  that maximizes  $I(X_1^n|\theta)$ ? (The achieving prior, same as in channel capacity.)

$$\max_w \min_q (\bar{R}(q, w)) = \max_w I(X_1^n|\theta) = C$$

From  $w^*(\theta)$  we derive  $q^* = q_{w^*}$ .

- (4) Properties of the achieving prior  $w^*(\theta)$ :

$$D(p_{\theta} || q^*) = C, \text{ for } w^*(\theta) > 0$$

$$D(p_{\theta} || q^*) < C, \text{ for } w^*(\theta) = 0$$

(stems from Karush-Khun-Tucker).

For every prior  $w(\theta)$ ,

$$n \cdot R_{Min-Max} = \min_q \max_w F(q, w) \leq \max_w F(w, q^*) \leq C = n \cdot R_{Max-Min}$$

Which completes the proof that

$$R_{Min-Max} = R_{Max-Min} = C$$

which is known as the *redundancy-capacity* theorem for universal coding.

(\*) If the capacity does not grow slower than linearly, i.e. if  $\lim_{n \rightarrow \infty} \frac{1}{n} C_n \neq 0$ , then universal coding is not possible.

Newer results have been found since:

- The strong version of the redundancy-capacity theorem[5, 6]: For every encoder (not necessarily  $q^*$ ), for *most* of the sources the redundancy is  $\geq C$ , that is, that the family of 'bad' parameters  $\Theta^*$  consists of almost all the sources  $\theta \in \Theta$ .
- For a smooth family with  $K$  parameters, it was further shown [6, 5], that  $C_n \cong \frac{K}{2} \log(n)$ .

## REFERENCES

- [1] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & sons, 1991.
- [2] I. Csiszár and J. Körner. *Information Theory: Coding Theorems for Discrete Memoryless Systems*. Academic Press, 1981.
- [3] Lee D. Davisson. Universal noiseless coding. IT-19(6):783–795, November 1973.
- [4] R. E. Krichevski and V. E. Trofimov. The performance of universal encoding. *IEEE Trans. Inform. Theory*, IT-27:199–207, March 1981.
- [5] Neri Merhav and Meir Feder. A strong version of the redundancy-capacity theorem of universal coding. *IEEE Transactions on Information Theory*, 41(3):714–722, 1995.
- [6] Jorma Rissanen. Universal coding, information, prediction, and estimation. *IEEE Transactions on Information Theory*, 30(4):629–636, 1984.

דחיסת נתונים ואותות, סמסטר ב' 2005, פרופ' רמי זמיר, טל פילוסוף.

## דחיסה של סדרה בודדת – אלגוריתם הדחיסה של למפל – זיו

סוכם ע"י ניר וינברגר

זכור, ישנם שני מודלים עבור קידוד סדרה שפילוגה אינו ידוע:  
(1) מודל סטוכסטי, שבו מניחים שהסדרה נוצרה מתוך פילוג השייך למשפחה נתונה של פילוגים.  
(2) מודל דטרמיניסטי, שבו לא מניחים פילוג כלשהוא על המקור, ומנסים לקודד בצורה הטובה ביותר סדרה זו. גישה זו נקראת גם גישת סדרה בודדת (individual sequence).  
אלגוריתם הדחיסה של למפל – זיו (Ziv – Lempel) הוא אלגוריתם דחיסה לסדרה בודדת, המבוסס על פיסוק (parsing) של סדרת הכניסה למקודד.  
בקודים מבוססי פיסוק המקודד מחלק את סדרת הכניסה לפסקאות (יכוננו גם מילות מקור, מחרוזות, strings, phrases) ומוציא מילת קוד לכל פסקה. מכאן, שהמקודד מכיל מילון המכיל את כל מחרוזות הכניסה האפשריות לפיסוק, ומילת הקוד עבורן. באלגוריתמים אוניברסליים מילון זה יגדל בצורה מסתגלת, ככל שהמקודד יראה חלק גדול יותר של הסדרה.  
בד"כ נוה לתאר מילון של קוד פיסוק באמצעות עץ, כאשר עלי העץ הם מחרוזות הכניסה שהמקודד מכיר.

דוגמאות לקודים בשיטת פיסוק:

1. קוד Tunstall. זהו קוד אופטימלי הפועל בשיטת Variable to Block (V2B), בדיעת פילוג המקור. בשיטת קידוד זו קצב הקידוד מוגדר כ  $R = \frac{k}{E[L]} \left( \frac{\text{bits}}{\text{sample}} \right)$ , כאשר k הוא מס' הביטים הקבוע שיוצא מהמקודד לכל מחרוזת מקור בכניסה, ו  $E[L]$  היא התוחלת של אורך מילת המקור המפוסקת. בכדי להשיג קצב קידוד מינימלי נרצה להגדיל את גודל מחרוזות המקור המפוסקות.  
קוד זה נבנה בצורה דואלית לקוד האפמן:  
- מתחילים עם עץ בסיסי המכיל רק את א"ב המקור, כעלי העץ.  
- עד שמגיעים ל  $2^k$  עלים מחלקים את העלה בעל ההסתברות הגבוהה ביותר.  
ניתן להוכיח אופטימליות עבור קוד זה, כלומר  $R \xrightarrow{k \rightarrow \infty} H$ , כאשר H היא האנטרופיה.

- דוגמא: קידוד מקור Ber(p) ב 3 ביטים למחרוזת, כאשר p=0.3.  
נסמן T – עלי עץ המילון של הקוד. P – ההסתברויות של אותם עלים, בהתאמה.  
- נתחיל עם  $T = \{0,1\}$  ו  $P = \{0.7,0.3\}$ . כעת לעלה '0' יש את ההסתברות הגבוהה ביותר ולכן נחלק אותו ונקבל את העלים '00' ו '01', כלומר  $T = \{00,01,1\}$  ו  $P = \{0.49,0.21,0.3\}$ .  
- העלה '00' הוא בעל ההסתברות הגבוהה ביותר ולכן נחלק אותו. נקבל  $T = \{000,001,01,1\}$ ,  $P = \{0.343,0.147,0.21,0.3\}$ .  
- נחלק את '000', נקבל:  $T = \{0000,0001,001,01,1\}$ ,  $P = \{0.2401,0.1029,0.147,0.21,0.3\}$ .  
- נחלק את '1' נקבל:  $T = \{0000,0001,001,01,10,11\}$  ו  $P = \{0.2401,0.1029,0.147,0.21,0.21,0.09\}$ .  
- נחלק את '0000' ונקבל:  $T = \{00000,00001,0001,01,10,11\}$ ,  $P = \{0.16807,0.07203,0.1029,0.147,0.21,0.21,0.09\}$ .  
- ולבסוף נחלק את '01' ונקבל את העץ הסופי  $T = \{00000,00001,0001,010,011,10,11\}$ ,  $P = \{0.16807,0.07203,0.1029,0.147,0.147,0.063,0.21,0.09\}$ .

האורך הממוצע של פסקת מקור הוא  $E[L] = 3.2831$ , ולכן הקצב הוא  $R = \frac{3}{3.2831} = 0.913 \left( \frac{\text{bits}}{\text{sample}} \right)$  (האנטרופיה של המקור היא  $H = 0.881(\text{bits})$ ).

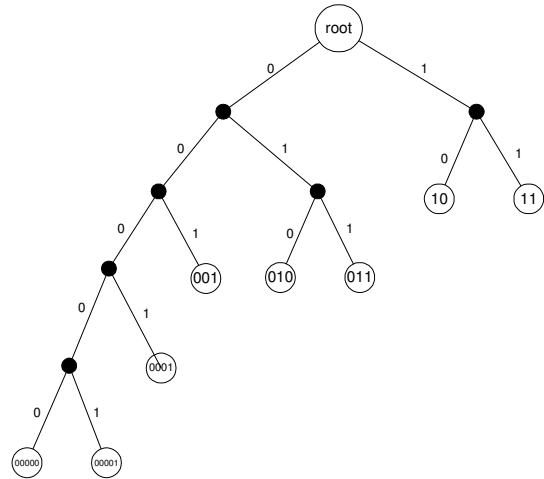


Figure 1 : Tunstall Tree

מכיון שהקידוד הוא למספר ביטים קבוע, נרצה שההסתברויות של כל המחרוזות אותן אנו מקודדים תהיה בערך שווה, וזאת משום שבאופן אופטימלי נרצה לקודד מחרוזות  $x^n$  בעלת הסתברות  $P(x^n)$  לאורך  $-\log_2 P(x^n)$  (כאן  $n$  הוא אורך מחרוזת המקור, ואורך זה משתנה). לכן, במילון הקוד נצפה למצוא מחרוזות המכילות רצפים ארוכים של סימבולי מקור בעלי הסתברות גבוהה, ולעומתם מחרוזות עם רצפים קצרים של סימבולי מקור בעלי הסתברות נמוכה. בדוגמא אכן ראינו שמופיעים בעץ רצפים ארוכים של הסימבול הסביר יותר '0' ורצפים קצרים יותר של '1'. כמו כן ההסתברויות שהתקבלו בעץ המילון היו בערך שוות, והאלגוריתם אכן "שואף" להשוות אותן ע"י חלוקת סימבולים בעלי הסתברות גבוהה מדי. קוד זה משפר את ההפסד הקיים בקודים אחרים באותיות הסבירות: מכיון ש  $P(x^n)$  הוא גבוה, הרי ש  $-\log_2 P(x^n)$  קטן בהרבה מ 1, ואם נעגל גודל זה למעלה  $\lceil -\log_2 P(x^n) \rceil = 1$  נפסיד הרבה באורך הממוצע.

2. Run Length Coding. בקוד זה נשדר את מספר הפעמים שהופיע הסימבול '0' ברצף לפני שהופיע הסימבול '1'. כמובן שקוד זה יפעל טוב כאשר המקור הוא בעל הסתברות גבוהה לקבלת '0'. לאחר שלב קידוד זה (הנותן קוד V2B), מכיון שמספר ה '0' לפני ה '1' הבא אינו בעל התפלגות אחידה, ניתן להוסיף קוד האפמן (או קוד B2V כלשהוא), ולקבל בסה"כ קוד V2V.

דוגמא: פיסוק הסדרה הבינארית ..00010110000010100100100010011. לפי Run Length Coding יתן  $3, 1, 0, 5, 1, 2, 2, 3, 2, 0, \dots$  שיקודדו ל  $0001, 01, 1, 000001, 01, 001, 001, 0001, 001, 1, \dots$ . לאחר מכן ניתן גם לקודד את "מספר האפסים לפני '1'".

- יתרון קודי פיסוק ביהס לקוד האפמן: אם המקור הוא בעל גודל א"ב גדול, הרי שבקוד האפמן נצטרך לשמור טבלה אכספוננציאלית בגודל הבלוק, בעוד שבקודי פיסוק גודל בלוק אותיות המקור משתנה בהתאם להסתברות וזיכרון.

## גישות לקידוד סדרה בודדת (Individual Sequence)

1. הנחת מודל פרמטרי  $\{P_\theta(x^n), \theta \in \Theta\}$ , וקידוד לפי ה  $\theta$  הטוב ביותר עבור משפחה זו.

ראינו כי הפילוג האוניברסלי המתקבל בשיטה זו הוא  $P_u(x^n) = \frac{1}{M} \sum_{\theta=1}^M P_\theta(x^n)$ , כאשר  $|\Theta| = M$  (כלומר

הנחת התפלגות אחידה על מרחב הפרמטרים). היתירות במקרה זה היא  $\frac{1}{n} \lceil \log M \rceil$ , והיא שואפת לאפס אם  $M$  הוא פונקציה תת-אקספוננציאלית של  $n$ .

2. Kolmogorov Complexity (1965) – סיבוכיות של סדרה בודדת היא אורך תוכנית המחשב הבינארית הקצרה ביותר שמגדירה את הסדרה. קצב הקוד המתקבל הוא אורך התוכנית לחלק לאורך הסדרה. ניתן להוכיח שעבור סדרה אינסופית לעולם לא ניתן להחליט בוודאות אם כבר מצאנו את התיאור הקצר ביותר שלה. גישה זו לא מביאה לפתרון קונקרטי לבעיית הדחיסה האוניברסלית.

3. Finite State Machine Complexity (FSM) – בגישה זו נגדיר משפחה של מקודדים המכילים מס' מצבים סופי, ונמצא את המקודד הטוב ביותר עבור הסדרה.

נסמן:  $\underline{b}_n$  - מילת הקוד,  $x_n \in \mathcal{X}$  - אות המקור מתוך א"ב המקור,  $s_n \in S$  - המצב הנוכחי מתוך מרחב המצבים.

$$\begin{array}{l} \text{המקודד :} \quad \underline{b}_n = f(x_n, s_n) \\ \text{והמפענח :} \quad x_n = f^{-1}(\underline{b}_n, s_n) \\ s_{n+1} = g(x_n, s_n) \end{array}$$

מכיון שהדחיסה היא ללא עיוות (lossless) גם המפענח יודע את המצב במדויק, באיחור של צעד אחד אחרי המקודד.

נתמקד בסדרה אחת נתונה: ראשית נניח שפונקציית המעבר ממצב למצב,  $g(\cdot)$ , הוגדרה. מכיון שגם הסדרה ידועה הרי שהמצבים  $s_1, s_2, \dots, s_n$  ידועים ולכן כל שנותר הוא למצוא את הפונקציה  $f(\cdot)$  שתביא לאורך קוד מינימלי עבור סדרה זו.

מכיון שאנו יכולים לקודד באופן שונה עבור כל מצב הרי שבכדי לקודד באופן אופטימלי עבור סדרה זו נרצה למצוא את השכיחות של סימבולי המקור בכל מצב ומצב.

$$\text{נסמן : } n(s) - \text{מס' ההופעות של מצב } s \text{ בסדרה, כלומר } \sum_s n(s) = n$$

$$\sum_x n(x|s) = n(s) \text{ כלומר } n(x|s) \text{ מס' ההופעות של סימבול מקור } x \text{ כאשר המקודד במצב } s$$

מכאן, שהשכיחות של סימבולי המקור, בהנתן שהמקודד במצב  $s$ , היא  $\frac{n(x|s)}{n(s)}$ . לאחר שמקבלים שכיחותות

$$\text{אלו הפתרון האופטימלי הוא הקצאת אורך של } -\log_2 \frac{n(x|s)}{n(s)} \text{ לסימבול } x \text{ כאשר מצב המקודד הוא } s.$$

האורך הכולל של סדרת הקוד המתקבל הוא:

$$L(x^n | g) = \sum_{s=1}^{|S|} n(s) \sum_x \frac{n(x|s)}{n(s)} (-\log_2 \frac{n(x|s)}{n(s)}) = \sum_{s=1}^{|S|} n(s) \hat{H}(x^n | s)$$

$$\text{כאשר } \hat{H}(x^n | s) = -\sum_x \frac{n(x|s)}{n(s)} \log_2 \frac{n(x|s)}{n(s)} \text{ היא האנטרופיה האמפירית של הסדרה בהנתן מצב } s.$$

נרצה כמובן למצוא את פונקציית המעבר ממצב למצב,  $g(\cdot)$ , הטובה ביותר עבור הסדרה הנתונה, תוך שמירה על האילוץ של  $|S|$  מצבים. לפיכך, כאשר אורך הסדרה גדל לאינסוף הקצב הטוב ביותר שניתן להשיג הוא

$$\limsup_{n \rightarrow \infty} \frac{1}{n} \min_g L(x^n | g)$$

(ראה הגדרת  $\limsup$  בנספח).

$$\rho_{FSM} = \inf_{|S|} \limsup_{n \rightarrow \infty} \frac{1}{n} \min_g L(x^n | g, |S|) : |S| \text{ המצבים על מס' המצבים}$$

קצב FSM של הסדרה יוגדר כאינפימום על מס' המצבים  $|S|$  : ורק אחר כך הרשינו למס' המצבים להיות אינסופי (מכיוון שהקצב מונטוני יורד ב  $|S|$ , אינפימום על מס' המצבים  $|S|$  הוא השאפה של  $|S|$  לאינסוף).

אם היינו מבצעים זאת בסדר הפוך הרי שהפתרון (הטריויאלי) היה הקצאת מצב לכל אות מקור, והקצב המתקבל היה 0, לכל n.

הערות:

(1) אם הסדרה  $x^n$  נפלטת ממקור ארגודי עם קצב אנטרופיה H נקבל  $\rho_{FSM}(x^n) = H$  בהסתברות 1 (with probability one).

(2)  $\rho_{FSM}(x^\infty)$  יהיה האנטרופיה האמפירית של הסדרה.

(3) ניתן להשיג את קצב הקידוד  $\rho_{FSM}(x^\infty)$  ע"י קידוד לפי אלגוריתם Lempel - Ziv לכל סידרה.

## LZ אלגוריתמי

1. LZ77 : מוצא את ההתאמה הארוכה ביותר של סימבולי מקור חדשים לעבר, עם או ללא "tail biting".
2. LZ78 : מוצא את ההתאמה הארוכה ביותר של סימבולי מקור חדשים לפסקה שהיתה בעבר.
3. LZW : וריאציה על LZ78.

- למרות שכל אלגוריתמי LZ מתכנסים אסימפטוטית לאנטרופיה של המקור, הרי שעבור סדרות באורך "בינוני" (קובץ טקסט לדוגמא) יכולה להיות עדיפות לאלגוריתם אחד על פני האחרים.
- העיקרון הכללי : בניה מסתגלת של מילון הולך וגדל של מילים אפשריות שנבנית עם פיסוק הסדרה.
- הגדרות וסימונים :

n - אורך סדרת המקור (שהמקודד כבר קבל).

$c_n$  - מספר הפסקאות עד n.

$l(i)$  - אורך פסקה i,  $i = 1, \dots, c_n$ .

$M(i)$  - מספר המילים במילון בזמן שמקודדת פסקה i,  $i = 1, \dots, c_n$ .

$L_{code}(i)$  - אורך הקוד לפסקה ה i. מתקיים :  $L_{code}(i) = \lceil \log_2 M(i) \rceil$ .

$n(i)$  - מקום סיום פסקה i.

כמו כן מתקיים :  $n = \sum_{i=1}^{c_n} l(i)$  ו-  $\rho_{LZ}(x^n) = \frac{1}{n} \sum_{i=1}^{c_n} L_{code}(i)$ , כאשר  $\rho_{LZ}(x^n)$  הוא קצב הקוד עבור

הסדרה.

: LZ77 .1

הפסקה החדשה שניצור היא ההתאמה הארוכה ביותר שניתן לראות מאינדקס  $n+1$  והלאה לעבר, בתוספת אות מקור נוספת.  
 כלומר, הפסקה ה  $c_n + 1$  היא  $(x_{k+1}, \dots, x_{k+l-1}) = \max_{l: l \leq L} (x_{n+1}, \dots, x_{n+l-1})$  כך ש  $k \leq n-l$ , כאשר לא מרשים "tail biting" ו  $k < n$  כאשר מרשים זאת.  
 מכאן שבמקרה של "tail biting" ההתאמה תתחיל בעבר של הסדרה, אבל היא עלולה לכלול גם את תחילת הפסקה המקודדת. באופן כללי גרסת "tail biting" תתן פסקאות ארוכות יותר.

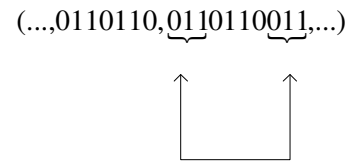


Figure 2 : tail biting

כמו כן, מבחינה מעשית הגבלנו את החיפוש בגודל ההתאמה ל 1 עד L בלבד.

פעולת הקידוד : המקודד צריך לשלוח את האינדקס שבו ההתאמה התחילה-  $k$ , את אורך ההתאמה -  $l$ , ואת החידוש -  $x_{n+l}$ . בסה"כ אורך הקוד המתקבל -  $|code(c_n)| = \lceil \log_2 n \rceil + \lceil \log_2 L \rceil + \lceil \log_2 |\chi| \rceil$ .

ניתן להראות שעבור מקור עם קצב אנטרופיה  $H$  מתקיים אופיינית  $l \approx \frac{\log n}{H}$  (ראה הצדקה בנספח) וכך

נבחר את  $L$ . לכן הקצב המתקבל הוא

$$\rho_{LZ}(x^n) = \frac{1}{n} \sum_{i=1}^{c_n} L_{code}(i) = \frac{\sum_{i=1}^{c_n} L_{code}(i)}{\sum_{i=1}^{c_n} l(i)} \approx \frac{\sum_{i=1}^{c_n} \log_2 n(i) + \log_2 \left( \frac{\log_2 n(i)}{H} \right)}{\sum_{i=1}^{c_n} \frac{\log_2 n(i)}{H}} \approx H$$

**דוגמא:** פיסוק הסדרה הבינארית ..00010110000010100100100010011. לפי LZ77 ללא "tail biting" יתן  $0,00,1,011,0000,01010,0100,10001,0011, \dots$ . עבור סדרה באורך 29 ביטים קיבלנו 9 פסקאות. כאשר נרשה "tail biting" נקבל  $0,001,011,0000,01010,01001000,10011$  כלומר 7 פסקאות. בדוגמא קיבלנו "tail biting" בקידוד הפסקה השניה והשישית.

: LZ78 .2

הפסקה החדשה שניצר היא ההתאמה הארוכה ביותר שניתן לראות מאינדקס  $n+1$  והלאה לפסקה שהיתה כבר בעבר, בתוספת אות מקור נוספת.

$$find : \max_l (x_{n+1}, \dots, x_{n+l}) = ( \underbrace{x_{n(i)+1}, \dots, x_{n(i)+l-1}}_{\text{existing word in the dictionary}}, \underbrace{x_{n+l}}_{\text{innovation}} )$$

פעולת הקידוד : המקודד צריך לשלוח אינדקס המפנה לפסקה המתאימה בעבר, ואת החידוש -  $x_{n+l}$ .



בסה"כ אורך הקוד המתקבל -  $L = \lceil \log_2(c_n + 1) \rceil + \lceil \log_2|\chi| \rceil$ , והקצב המתקבל בהזנחת תרומת אות

$$\rho_{LZ}(x^n) = \frac{1}{n} \sum_{i=1}^{c_n} L_{code}(i) \approx \frac{\log_2(c_n + 1)!}{n} \approx \frac{c_n \log c_n}{n}$$

החידוש וקירוב סטירלינג הוא

מכיון שבשיטת LZ78 מחפשים התאמות רק לפסקאות שהיו כבר ולא לרצף אותיות כלשהוא, נקבל שהפסקאות בשיטה זו קצרות יותר ( כלומר, "קצב פסיקים" גדול יותר) מאשר LZ77, אבל מצד שני גודל המילון קטן יותר ולכן מילות הקוד המתאימות לפסקאות אלו קצרות יותר.

דוגמא: פיסוק הסדרה הבינארית 00010110000010100100100010011.. לפי LZ78 יתן 0,00,1,01,10,000,010,100,1001,0001,001,1... (זה '1' האחרון יצטרף לפסקה הבאה).

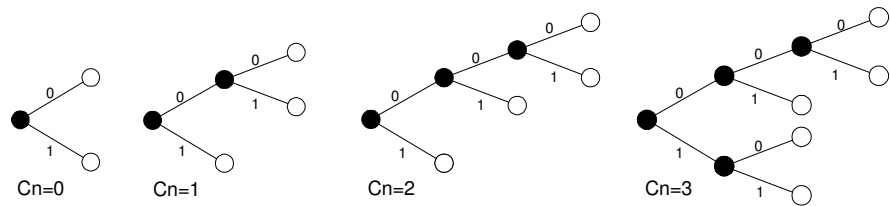


Figure 3 : An adaptively growing LZ78 tree

- מה היא הסדרה הגרועה ביותר עבור LZ78 ?
- סדרת המניה (Counting series) הנתונה ע"י : 0,1,00,01,10,11,000,001,010,011,... (הפסיקים לא מופיעים בסדרה המקורית, אלא מתקבלים לאחר הפיסוק של LZ78). המחשה : אורך הפסקה לעומת מספרה –

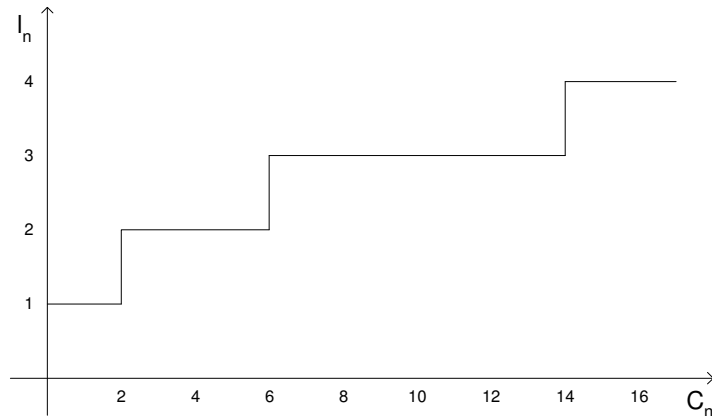


Figure 4 : Phrases length for counting series

לאחר שנסתיים מחזור מניה שכל פסקה בו היא באורך L ביטים נקבל שמס' הפסקאות -  $c = \sum_{l=1}^L 2^l$

ואורך הסדרה הכולל -  $n = \sum_{l=1}^L l \cdot 2^l$ . מכאן ניתן לקבל שמס' הפסקאות עד אורך n הוא  $c_n \approx \frac{n}{\log n}$

ולכן  $L \approx L_n \approx \frac{n}{c_n} \approx \log n$ . מתקבל  $1 - O\left(\frac{\log \log n}{\log n}\right) \approx 1$ .  
 $\rho_{LZ}(x^n) \approx \frac{c_n \log c_n}{n} \approx \frac{\frac{n}{\log n} \cdot \log\left(\frac{n}{\log n}\right)}{n} = 1 - O\left(\frac{\log \log n}{\log n}\right) \approx 1$ .  
 כלומר הסדרה לא ניתנת לדחיסה.

- "סדרת מניה למקור ארגודי": כעת, בכל אורך  $l$  יופיעו רק סדרות אופייניות למקור, ולכן נקבל בערך  $2^{lH}$  סדרות אפשריות (מתוך ה AEP). בכדי שאלגוריתם LZ78 אכן יפסק כל סדרה כזו, נניח לצורך הדיון שסדרה אופיינית באורך  $l+1$  היא הארכה של סדרה אופיינית באורך  $l$  (נכון בקירוב עבור  $l$  גדול). נקבל שמש' הפסקאות הוא  $c = \sum_{l=1}^L 2^{lH}$ , ואורך הסדרה הכולל הוא  $n = \sum_{l=1}^L l \cdot 2^{lH}$ . ניתן להראות בצורה דומה כי  $c_n \approx \frac{n}{\log n} H$ ,  $L_n = \frac{\log n}{H}$ , ו  $\rho_{LZ}(x^n) \approx \frac{c_n \cdot \log c_n}{n} \approx H$ . מכאן שאלגוריתם LZ מגיע אסימפטוטית לאנטרופיה.

### השגת האנטרופיה האמפירית של סדרה בודדת ע"י LZ78

הגדרה: אנטרופיה אמפירית מסדר  $l$  של הסדרה  $x^n$ :  $\hat{H}_l(x^n) \triangleq - \sum_{\underline{x} \in \mathcal{X}^l} \hat{P}_l(\underline{x}) \log_2 \hat{P}_l(\underline{x})$ . כאשר

$$\hat{P}_l(\underline{x}) \text{ הוא שערך של ההסתברות ב"חלון נע" } \hat{P}_l(\underline{x}) = \frac{n(\underline{x} | x^n)}{n-l+1} \text{ מתוך הסדרה } x^n.$$

לסדרה אינסופית ניקח את הגבול  $\hat{H}_l(x^\infty) = \limsup_{n \rightarrow \infty} \hat{H}_l(x^n)$ , ועבור מקור סטציונרי או ארגודי נקבל

$$\hat{H}_l(x^\infty) \xrightarrow{l \rightarrow \infty} H \text{ (with probability one) בהסתברות 1}$$

$$\forall l, \rho_{LZ}(x^n) \leq \hat{H}_l(x^n) + O\left(\frac{\log \log n}{\log n}\right) \text{ מקיים LZ78}$$

הוכחה: נוכיח רק שעבור  $l=1$ , ומקור בינארי מתקיים  $\rho_{LZ}(x^n) \leq h_B(\hat{P}_1(x^n)) + \frac{c_n \log\left(\frac{n}{c_n}\right)}{2n}$ , כאשר

$h_B(\cdot)$  היא פונקציית האנטרופיה הבינארית ו  $\hat{P}_1(\underline{x})$  היא ההסתברות האמפירית מסדר ראשון של הסדרה.

כמו כן נשים לב כי  $\frac{n}{c_n}$  הוא אורך ממוצע של פסקה (והוא פרופורציוני ל  $O(\log n)$ ).

מכיון שכאשר מקודדים פסקה מסוימת נותנים הסתברות שווה לכל המילים במילון (כלומר אורך שווה) ניתן לומר שפעולת קידוד זו משרה הסתברויות שוות על עץ המילון. כמו כן פעולה זו משרה הסתברויות על הענפים שמובילים לעלים. נגדיר הסתברות של ענף כהסתברות לעבור באותו ענף, בהנתן שאנו נמצאים בצומת שהוא מתחיל בו. נקבל:

$$\Pr(\text{branch}) \equiv \frac{\text{number of leaves connected to the end of the branch}}{\text{number of leaves connected to the beginning of the branch}}$$

$$\text{ומתקיים } \Pr(\text{leaf}) \equiv \prod_{\text{all the branches from root to leaf}} \Pr(\text{branch}) = \frac{1}{M}$$

$$L(\text{leaf}) = \log_2 M = -\log_2 \Pr(\text{leaf}) = - \sum_{\text{all the branches from root to leaf}} \log_2 \Pr(\text{branch})$$

נתיחס לדוגמא לסדרה (0,01,010,1,0100,00) (מפוסקת, כמובן, באמצעות LZ78). נסתכל על עץ הקוד וההסתברויות המושרות על הענפים בשלבים שונים של הסדרה:

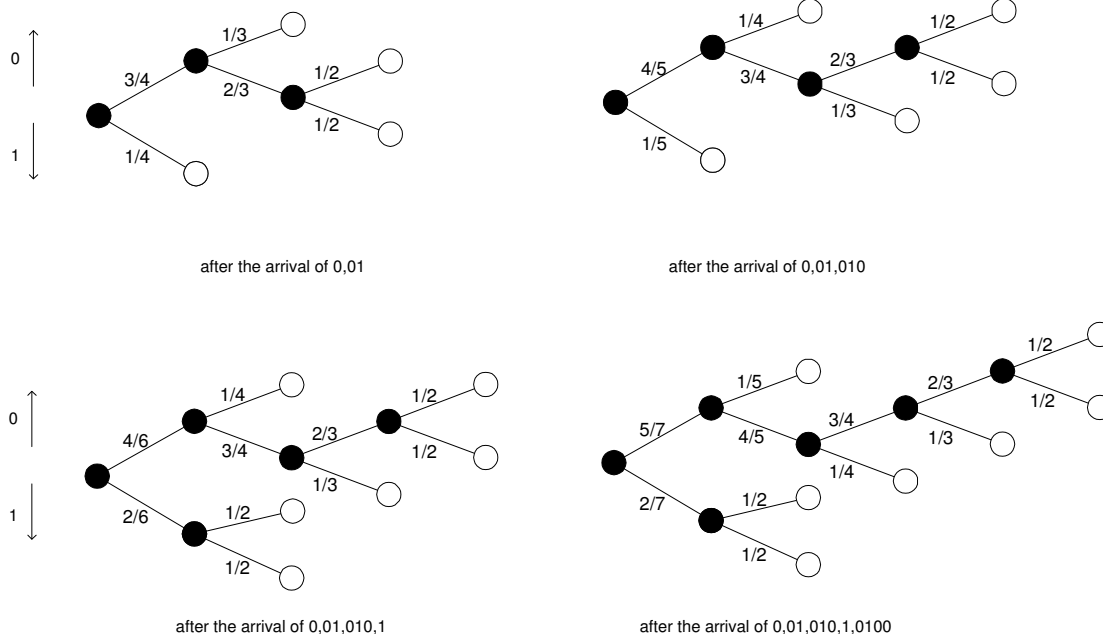


Figure 5 :Probability assignment for branches in a LZ78 tree

אורך הקוד הכולל עבור סדרה זו הוא כמובן סכום של אורכי הקוד עבור כל מילה, כלומר

$$L(\text{code}) = \sum_{\text{coded leaves}} - \sum_{\text{all the branches from root to leaf}} \log_2 \Pr(\text{branch}) .$$

נרשום עבור הדוגמא את ההסתברות של כל מילה במילון כמכפלת ההסתברויות הענפים (לפי ההסתברויות שהושרו עליהם כאשר מילה זו קודדה):

PHRASE	*	*□	*□	*	*□	*□
	0	01	010	1	0100	00
$\prod \Pr(\text{branch})$	$\frac{1}{2}$	$\frac{2}{3} \cdot \frac{1}{2}$	$\frac{3}{4} \cdot \frac{2}{3} \cdot \frac{1}{2}$	$\frac{1}{5}$	$\frac{4}{6} \cdot \frac{3}{4} \cdot \frac{2}{3} \cdot \frac{1}{2}$	$\frac{5}{7} \cdot \frac{1}{5}$
Pr(leaf)	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{5}$	$\frac{1}{6}$	$\frac{1}{7}$

ואכן  $\Pr(\text{leaf}) = \prod \Pr(\text{branch})$ .

כעת נשנה את סדר הסכימה עבור אורך הקוד: במקום לסכום על מסלול בכל עץ ואז לסכום על כל העצים, נסכום על צומת אחד בכל העצים, ואז נסכום על כל הצמתים (שמסלול כלשהוא עבר בהם, באחד העצים). מכיון שאנחנו סוכמים אורכים, הדבר שקול להכפלת ההסתברויות. נשים לב שהשורש מיצג את תחילת הפסקאות, הצמתים שמעליו מיצגים את האות השניה בפסקאות, וכן הלאה עד העלים, המיצגים את סוף הפסקאות.

נתבונן בצומת כלשהוא ובאחד הענפים היוצא ממנו. בכל פעם שנעבור בצומת ההסתברות של הענף תשתנה. בכל מקרה, כאשר נעבור בצומת, יתווסף עלה המחובר לצומת זו (שממנה יצא הענף), ולכן המכנה של הסתברות הענף יגדל ב 1. מכיון שבפעם הראשונה שנעבור בצומת, יתוספו אליה שני עלים, ובכל פעם אחרת יתווסף עלה בודד (למעשה יתוספו שני עלים, אבל העלה שהעלים החדשים יצאו ממנו לא יהיה יותר עלה), נקבל שאם עברנו בצומת  $i$  פעמים, כלומר  $i$  ביטים יצאו ממנה, מחוברים אליה  $i+2$  עלים, וזהו המכנה של הסתברות הענף. המונה של הסתברות הענף יתחיל מ 1, כאשר עדיין לא עברנו בצומת בכלל, ויגדל בהתאם לביטים שיוצאים מהצומת. אם הביט שהגיע הוא הביט שהענף מיצג (כלומר המשכנו בכיוון הענף) יתווסף עלה לצומת שאליה נכנס הענף, ולכן המונה יגדל ב 1, אחרת, הוא ישאר אותו דבר. מכאן שכאשר מגיע הביט

ה  $i+1$  שיוצא מהצומת ההסתברות של הענף היא  $P_i(b) \triangleq \frac{n_b(\tilde{x}_1^i) + 1}{i + 2}$ , כאשר  $\tilde{x}_1^i$  הם  $i$  הביטים שיצאו

מצומת זו מקודם,  $b$  הוא '0' או '1' ו  $n_b(\tilde{x}_1^i)$  הוא מס' ההופעות של ביט  $b$  בוקטור  $\tilde{x}_1^i$ . ניתן לראות שזהו בדיוק שערך לפלאס להסתברות מתוך סדרת הביטים שיצאה מהצומת בכל פעם שהיינו בה  $\tilde{x}_1^i$ . בסה"כ, כאשר נכפיל את ההסתברויות של הענפים שיצאו מהצומת בכל העצים, נקבל את הפילוג האוניברסלי השקול

$$P_u(\tilde{x}_1^n) \triangleq \frac{n_0(\tilde{x}_1^n)! n_1(\tilde{x}_1^n)!}{(n+1)!}, \text{ והראינו שמתקבל}$$

נאשר זאת ע"י הדוגמא. נסתכל על צומת השורש. בסדרה יש 6 מסלולים היוצאים מצומת זה, והביטים שיוצאים מהשורש הם 00100 (מסומנים ב \*). לפי הטבלה, ההסתברות השקולה שלהם היא:

$$\frac{1}{2} \cdot \frac{2}{3} \cdot \frac{3}{4} \cdot \frac{1}{5} \cdot \frac{4}{6} \cdot \frac{5}{7} = \frac{5! \cdot 1!}{(6+1)!} = P_u(00100)$$

נסתכל גם על הצומת  $0_0$ , כלומר האות השנייה בכל הפסקאות שמתחילות באות 0. בסדרה יש 4 מסלולים העוברים דרך צומת זה, והביטים שיוצאים הם 1110 (מסומנים ב □).

ההסתברות השקולה שלהם היא:  $P_u(1110) = \frac{3! \cdot 1!}{(4+1)!} = \frac{1}{2} \cdot \frac{2}{3} \cdot \frac{3}{4} \cdot \frac{1}{5}$ . גם כאן קיבלנו את שיערוך לפלאס

עבור הסדרה היוצאת מצומת זה.

נגדיר: לאחר פיסוק וקידוד  $c_n$  פסקאות -

$n_i$  - מס' המעברים בצומת  $i$ ,  $i=1..c_n$  (משום שכל פסקה חדשה מוסיפה צומת פנימי חדש לעץ, ולכן מס' הפסקאות הוא כמספר הצמתים).

כמו כן מתקיים  $n = \sum_{i=1}^{c_n} n_i$  וזאת משום שכל מעבר בצומת בעץ משמעותו

אות מקור.

$\tilde{x}_i$  - וקטור אותיות המקור המתאים למעברים מצומת  $i$ .

$\hat{P}_1(\tilde{x}_i)$  - ההסתברות האמפירית של  $\tilde{x}_i$  מסדר ראשון ל '1', הנתונה ע"י  $\frac{n_1(\tilde{x}_i)}{n_b(\tilde{x}_i)}$  הוא מס' ההופעות של ביט  $b$  בוקטור  $\tilde{x}_i$ .

כעת:

$$total \ code \ length = \sum_{\substack{\text{coded} \\ \text{leaves}}} - \sum_{\substack{\text{all the branches} \\ \text{from root to leaf}}} \log_2 \Pr(branch) = \sum_{\substack{\text{inner nodes} \\ \text{passes through} \\ \text{node}}} - \sum \log_2 \Pr(branch)$$

מכיון שסכום על המעברים בצומת נותן את הפילוג האוניברסלי, וישנם  $c_n$  צמתים פנימיים (כל פסקה מוסיפה צומת פנימי), נקבל:

$$total \ code \ length = -\sum_{i=1}^{c_n} \log_2 P_u(\tilde{x}_i) = -\sum_{i=1}^{c_n} \log_2 \frac{n_0(\tilde{x}_i)! n_1(\tilde{x}_i)!}{(n_i + 1)!}$$

בעזרת קירוב סטירלינג ניתן להראות כי  $-\log_2 P_u(\tilde{x}_i) \approx n_i \cdot h_B(\hat{P}_1(\tilde{x}_i)) + \frac{\log_2 n_i}{2}$  ולכן

$$\text{total code length} \approx \sum_{i=1}^{c_n} n_i \cdot h_B(\hat{P}_1(\tilde{x}_i)) + \frac{1}{2} \log_2 n_i$$

כלומר הקצב הוא

$$\rho_{LZ}(x^n) = \frac{\text{total code length}}{n} = \sum_{i=1}^{c_n} \frac{n_i}{n} \cdot h_B(\hat{P}_1(\tilde{x}_i)) + \frac{c_n}{2n} \sum_{i=1}^{c_n} \frac{1}{c_n} \log_2 n_i$$

ומתכונת הקעירות (concavity) של פונקציית האנטרופיה וה  $\log$  נקבל ע"י אי שיויון Jensen

$$\rho_{LZ}(x^n) \leq h_B\left(\sum_{i=1}^{c_n} \frac{n_i}{n} (\hat{P}_1(\tilde{x}_i))\right) + \frac{c_n}{2n} \log_2\left(\sum_{i=1}^{c_n} \frac{n_i}{c_n}\right)$$

קל לראות כי  $\sum_{i=1}^{c_n} \frac{n_i}{n} (\hat{P}_1(\tilde{x}_i))$  הוא ההסתברות האפירית של הסדרה כולה,  $\hat{P}_1(x^n)$ , (למעשה ממצעים את

ההסתברות האמפירית ל '1' בהנתן צומת כלשהוא, לפי השכיחות של כל צומת) ולכן

$$\rho_{LZ}(x^n) \leq h_B(\hat{P}_1(x^n)) + \frac{c_n}{2n} \log_2\left(\frac{n}{c_n}\right)$$

מ.ש.ל. □

מתי נקבל שיויון?

לפי התנאי לשיויון באי שיויון Jensen, בכדי שנשיג את הביטוי הראשון בשיויון דרוש כי כל  $0 < \theta < 1$ ,  $Ber(\theta)$ , כלומר  $\forall i$ . נקבל זאת בהסתברות גבוהה במקורות חסרי זיכרון, כלומר  $Ber(\theta)$ , כלשהוא, כאשר הפסקאות ארוכות מספיק. אם המקור בעל זיכרון נקבל בהסתברות גבוהה אי שיויון ממש, והקצב המתקבל יהיה קטן יותר מאשר האנטרופיה האמפירית מסדר ראשון, בתוספת היתירות.

הערות:

(1) לכל סדרה אינדיבידואלית נקבל  $\rho_{LZ}(x^\infty) \leq \hat{H}_\infty(x^\infty)$ .

(2) הוכח שלכל מכונת מצבים סופית מתקיים  $\rho_{FSM}(x^\infty) \geq \hat{H}_\infty(x^\infty)$ .

(3) מ (1) ו (2) נקבל  $\rho_{LZ}(x^\infty) = \rho_{FSM}(x^\infty) = \hat{H}_\infty(x^\infty)$  (נכון למרות ש LZ היא לא בדיוק מכונת מצבים סופית משום שמספר המצבים שלה גדל עם n).

(4) Savari (1997) הראתה שעבור מקורות ארגודיים היתירות של LZ78 היא בעצם  $O\left(\frac{1}{\log n}\right)$ , כלומר

אסימפטוטית היא קצת טובה יותר מהיתירות של LZ77.

(5) ניתן להרחיב את ההוכחה עבור  $l > 1$ , ולהראות ש LZ78 משיג את האנטרופיה האמפירית המרקובית מסדר k. נראה לדוגמא עבור מקור מרקובי מסדר ראשון:

ההסתברות האמפירית ל '1' של ביטים שיוצאים מצומת תהיה תלויה בביט שהופיע לפני הצומת. נחזור על אותה הוכחה עד לשלב בו קיבלנו

$$\rho_{LZ}(x^n) = \frac{\text{total code length}}{n} = \sum_{i=1}^{c_n} \frac{n_i}{n} \cdot h_B(\hat{P}_u(\tilde{x}_i)) + \frac{c_n}{2n} \sum_{i=1}^{c_n} \frac{1}{c_n} \log_2 n_i$$

כעת נפריד את הסכום על  $c_n$  הצמתים לצמתים שמופיע לפניהם '0' וצמתים שמופיע לפניהם '1':

$$\rho_{LZ}(x^n) = \sum_{\substack{\text{inner nodes} \\ \text{after '0'}}} \frac{n_i}{n} \cdot h_B(\hat{P}_u(\tilde{x}_i)) + \sum_{\substack{\text{inner nodes} \\ \text{after '1'}}} \frac{n_i}{n} \cdot h_B(\hat{P}_u(\tilde{x}_i)) + \frac{c_n}{2n} \sum_{i=1}^{c_n} \frac{1}{c_n} \log_2 n_i$$

נסמן את מספר הצמתים המופיעים לאחר '0' ו'1' כ  $n_0(x^n)$  ו  $n_1(x^n)$ , בהתאמה, ונקבל:

$$\rho_{LZ}(x^n) = \frac{n_0(x^n)}{n} \sum_{\substack{\text{inner nodes} \\ \text{after '0'}}} \frac{n_i}{n_0(x^n)} \cdot h_B(\hat{P}_u(\tilde{x}_i)) + \frac{n_1(x^n)}{n} \sum_{\substack{\text{inner nodes} \\ \text{after '1'}}} \frac{n_i}{n_1(x^n)} \cdot h_B(\hat{P}_u(\tilde{x}_i)) + \frac{c_n}{2n} \sum_{i=1}^{c_n} \frac{1}{c_n} \log_2 n_i$$

ולפי תכונת הקעירות של פונקציית האנטרופיה וה  $\log$  נקבל ע"י אי שיוויון Jensen בצורה דומה:

$$\rho_{LZ}(x^n) \leq \frac{n_0(x^n)}{n} h_B \left( \sum_{\substack{\text{inner nodes} \\ \text{after '0'}}} \frac{n_i}{n_0(x^n)} \hat{P}_1(\tilde{x}_i) \right) + \frac{n_1(x^n)}{n} h_B \left( \sum_{\substack{\text{inner nodes} \\ \text{after '1'}}} \frac{n_i}{n_1(x^n)} \hat{P}_1(\tilde{x}_i) \right) + \frac{c_n}{2n} \log_2 \left( \sum_{i=1}^{c_n} \frac{n_i}{c_n} \right)$$

כעת קל להוכיח כי  $\sum_{\substack{\text{inner nodes} \\ \text{after '0'}}} \frac{n_i}{n_0} \hat{P}_1(\tilde{x}_i)$  הוא ההסתברות האמפירית של '1' בהנתן שהביט הקודם הוא '0',

ו  $\sum_{\substack{\text{inner nodes} \\ \text{after '1'}}} \frac{n_i}{n_1} \hat{P}_1(\tilde{x}_i)$  הוא ההסתברות האמפירית של '1' בהנתן שהביט הקודם הוא '1'. כמו כן  $\frac{n_0(x^n)}{n}$

הוא ההסתברות האמפירית של '0' בסדרה (כלומר  $1 - \hat{P}_1(x^n)$ ) ו  $\frac{n_1(x^n)}{n}$  הוא ההסתברות האמפירית של

'0' בסדרה  $\hat{P}_1(x^n)$ . בסה"כ נקבל כי

$$\rho_{LZ}(x^n) \leq (1 - \hat{P}_1(x^n)) \cdot h_B(\hat{P}_1(x^n | \text{last bit was '0'})) + \hat{P}_1(x^n) \cdot h_B(\hat{P}_1(x^n | \text{last bit was '1'})) + \frac{c_n}{2n} \log_2 \left( \sum_{i=1}^{c_n} \frac{n_i}{c_n} \right)$$

ושני הביטויים הראשונים הם האנטרופיה המותנית בביט אחד אחורה, המחושבת אמפירית על סמך  $x^n$ ,

$$\cdot \rho_{LZ}(x^n) \leq \hat{H}(x_n | x_{n-1}) + \frac{c_n}{2n} \log_2 \left( \sum_{i=1}^{c_n} \frac{n_i}{c_n} \right) \text{ בסה"כ } \hat{H}(x_n | x_{n-1})$$

עבור מקור מרקובי כללי מסדר k נפריד בצורה דומה את הסכום על כל הצמתים הפנימיים לפי k הביטים

$$\cdot \rho_{LZ}(x^n) \leq \hat{H}(x_n | x_{n-k}^{n-1}) + \frac{c_n}{2n} \log_2 \left( \sum_{i=1}^{c_n} \frac{n_i}{c_n} \right) \text{ ונקבל}$$

דומה למקודם נקבל שיוויון כאשר מתקבל שיוויון באי שיוויון Jensen. בהסתברות גבוהה ועבור פסקאות מספיק ארוכות נקבל שיוויון כאשר המקור הוא מרקובי מסדר k. אם המקור מרקובי מסדר גבוה יותר, בהסתברות גבוהה הקצב יהיה נמוך יותר.

(6) גודל המקדם של  $O(\cdot)$  עולה עם גודל הא"ב לפי  $\frac{|\chi|-1}{2}$ , וגם עולה ככל שאורך הזיכרון של המקור גדול יותר.

### LZW .3 :

אלגוריתם זה דומה ל LZ78 למעט שינוי שהוסיף Welch (1984). שינוי זה נועד להתגבר על חוסר היעילות הנמצא באלגוריתמים הקודמים שהוצגו : אות המקור האחרונה של כל פסקה תמיד נשלחה לא מקודדת. הדבר חשוב בעיקר במקורות "דחיסים", כלומר מקורות שקצב האנטרופיה שלהם נמוך בהרבה מ  $\lceil \log_2 |\chi| \rceil$ . דוגמא נפוצה למקור כזה הוא קובץ טקסט. אם נפעיל את אלגוריתם LZ78 על מקור כזה הרי שלאחר כל פסקה נצטרך להוסיף למילון את כל האפשרויות לאות חידוש. אם גודל הא"ב גדול נצטרך להוסיף הרבה מילים חדשות למילון. האלגוריתם של Welch לא מוסיף בכל שלב את כל האפשרויות לאות חידוש, אלא רק אות אחת – האות שהופיעה אחרי הפסקה הנוכחית. כלומר האלגוריתם מקודד כך:

- הפסוק מבוצע ביחס למילון.
  - ברגע 0 המילון הוא הא"ב של המקור  $\chi$ .
  - בכל פיסוק מתווספת מילה חדשה (בודדת) למילון : הפסקה החדשה שמצאנו בתוספת האות הראשונה בפסקה הבאה בסדרה.
- מכאן שהפיסוק מחפש את ההתאמה הארוכה ביותר האפשרית: ל  $\chi$ , לפסקה קיימת, או לפסקה קיימת בתוספת האות אחת אחריה בסדרה.

#### הערות :

- (1) בעץ הקוד המילים הם לאו דוקא עלים. בד"כ ישנם גם צמתים פנימיים שהם מילים - אלו צמתים שעדיין לא קיבלו את כל האפשרויות לאות מקור אחריהן. נרצה, כמובן, למצוא את ההתאמה הארוכה ביותר ולא לעצור בצומת פנימי כאשר ניתן למצוא התאמה ארוכה יותר.
  - (2) מכאן, שכאשר המקודד יראה את כל האפשרויות לאות מקור אחרי פסקה כלשהיא נרצה לבטל את ההתאמה לפסקה זו, ולקחת מכאן והלאה רק התאמות ארוכות יותר ( באות אחת, לפחות).
  - (3) נשים לב שהאות הנוספת לא מקודדת בפסקה הנוכחית, והמפענח לא מכיר אותה. רק כאשר תשלח הפסקה הבאה והמפענח יראה את האות המופיעה אחרי הפסקה הנוכחית הוא יוכל להגדיל את המילון שלו. לכאורה, שיטת קידוד זו עלולה להיות בעייתית. מה יקרה כאשר המקודד יפנה לפסקה אחת לפני זאת שאנחנו רוצים לקודד, וירצה להוסיף לה את האות הראשונה של הפסקה הנוכחית? המפענח עדיין לא מכיר אות זאת משום שאין לו עדיין את הפסקה הנוכחית, ולא יוכל לפענח.
- בפועל, אין כאן רב משמעות משום שמצב זה יכול להיווצר רק כאשר הסדרה היא מהצורה  $..., aw, awa, ...$ , (  $a$  – אות מקור,  $w$  – וקטור אותיות מקור), ולכן בכל אופן האות הנוספת שיש להוסיף היא האות הראשונה של הפסקה, והמפענח יכול לדעת זאת גם ללא קבלת הפסקה הנוכחית.

דוגמא: פיסוק הסדרה הבינארית ..00010110000010100100100010011. לפי LZW יתן 0,00,1,0,1,10,00,001,01,0010,010,00100,11,... עבור סדרה באורך 29 ביטים נקבל בדיוק 13 פסקאות.

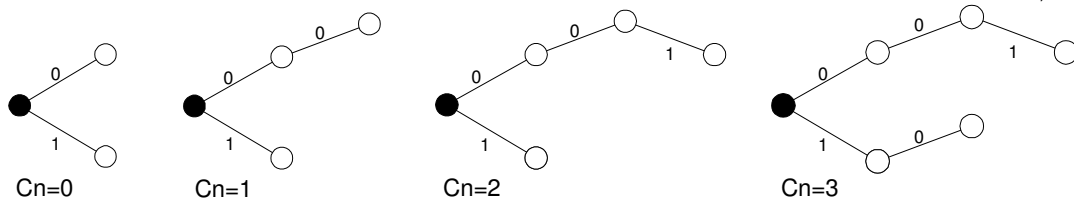


Figure 6 : An adaptively growing LZW tree

## קידוד מספר שלם

כיצד נקודד מספר שלם כלשהוא ללא הנחה הסתברותית?

מוטיבציה: ראינו כי היתירות בדחיסה של משפחה פרמטרית היא  $R \sim K \frac{1}{2n} \log_2 n$ , כאשר  $K$  הוא מס' הפרמטרים. לעומת גישה זו אלגוריתם LZ משיג אוניברסליות למרות שיש לו מס' אינסופי של פרמטרים. נרצה להשיג אוניברסליות בגישה פרמטרית גם כאשר סדר המודל אינו ידוע מראש. Rissanen הראה double universality (1984): נניח שיש משפחות פרמטריות  $\Theta_1, \Theta_2, \dots$  כאשר כל  $\Theta_i$  מיצג משפחה פרמטרית שונה (לדוגמא AR(i), או תחום פרמטרים הולך וגדל עם  $i$ ). במצב זה נרצה למצוא מקודד המביא למינימום את  $\min_{K, \theta} |code(K)| + |code(\theta | \Theta_K)| + |code(x^n | \theta)|$ . בכדי לקודד את סדר המודל נרצה לדעת כיצד ניתן לקודד מס' שלם כלשהוא.

הערה: Rissanen הציג זאת גם כעיקרון לבחירה נכונה של סדר המודל בבעיות שערוך, וקרא לו Minimum description length (MLD).

אם ידוע שהמספר  $n$  נמצא בתחום  $[0, N-1]$  נוכל לקודד ע"י  $\log_2 N$  ביטים. מכאן שעבור כל  $n$  בתחום זה מתקיים שאורך מילת הקוד לתיאור  $n$  גדול מ  $\log_2 n$ . Elias מצא שיטה (1975) שבה ניתן לקבל  $|code(n)| = \log^* n$  (ראה הגדרת  $\log^* n$  בנספח).

נדגים את פעולת האלגוריתם על קידוד המספרים בתחום  $[0, N-1]$ , כאשר  $N = 2^{2^2} = 2^{16} = 64K$ .  
 - בכדי לקודד מס' בתחום  $2^{15} \div 2^{16}$  דרושים 16 ביטים.  
 - בכדי לקודד מס' בתחום  $2^{14} \div 2^{15}$  דרושים 15 ביטים.  
 מילת הקוד עבור מספר כלשהוא בתחום תהיה מהצורה:

$$code(n) = [field\ 0] + [field\ 1] + [field\ 2] + [field\ 3]$$

כאשר שדה  $i$  הוא באורך אפשרי של  $2^{i+1} \div 1$ . בכדי להגיע למילת קוד קצרה לא נשתמש תמיד בכל השדות, וגם לא בכל אורך השדה האפשרי. בכדי להודיע זאת למפענח דרוש סימן נוסף, נסמנו ESC. מנגנון אפשרי ל ESC יכול להיות שבסוף כל שדה נוסף ביט נוסף, שכל תפקידו לומר למפענח האם זהו סוף התיאור או שיש שדה נוסף. עבור  $n$  גדול התוספת לאורך הכולל של מילת הקוד זניח.

פעולת הפיענוח תתבצע כך:

- אם אחר שדה מופיע הסימן ESC הרי שהשדה אומר את ערכו של המספר שקודד.
  - אם לאחר שדה מופיע שדה נוסף, השדה הנוכחי אומר את אורכו של השדה הבא.
- לכן, המקודד ישתמש בשדות שאומרים למפענח את גודל השדה הבא, עד שהשדה יהיה מספיק גדול לקידוד המספר.

דוגמא: קידוד  $2^{15} < 13500 + 2^{14}$ . דרושים 15 ביטים עבור קידוד מספר זה.

שדה 0 יאמר למקודד ששדה 1 הוא בגודל 2 ביט.

שדה 1 יאמר למקודד ששדה 2 הוא בגודל 4 ביט.

שדה 2 יאמר למקודד ששדה 3 הוא בגודל 15 ביט.

כעת, שדה 3 הוא מספיק גדול לקידוד המספר ולכן הוא יכול את הקוד עבור המספר.

נקבל:  $code(2^{14} + 13500) = [1] + [11] + [1110] + [111010010111100] + ESC$ , כאשר מספר

בשדה ללא ESC אחריי אומר שהשדה הבא הוא בגודל ערך המספר הבינארי ועוד 1, והשדה האחרון אומר את המספר במדויק.



## נספחים

1. הגדרת  $\limsup$ : עבור סדרת מספרים ממשית  $a_n$  נסמן  $A_n = \sup_{k \geq n} a_k$ , ונגדיר  $\limsup_{n \rightarrow \infty} a_n = \lim_{n \rightarrow \infty} A_n$ . גבול זה תמיד קיים משום ש  $A_n$  היא סדרה לא עולה – אם היא חסומה יש לה גבול סופי, אחרת גבולה  $-\infty$ . אם נגדיר גבול חלקי של  $a_n$  כגבול של סדרה חלקית של  $a_n$ , הרי ש  $\limsup_{n \rightarrow \infty} a_n$  הוא הגבול החלקי הגדול ביותר של  $a_n$ .

ניתן גם לחשוב על  $\limsup a_n$  בתור החסם העליון ההדוק (נמוך) ביותר לזנב הסדרה.

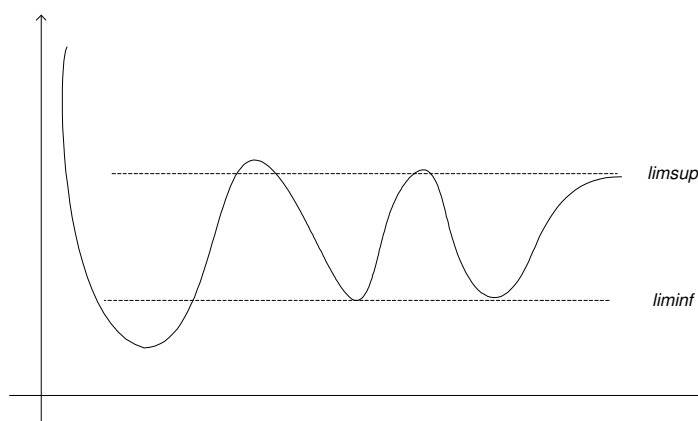


Figure 7 : limsup & liminf

2. הגדרת לוגריתם חוזר  $\log^* n$  - (iterated logarithm):

$$\log^* n \hat{=} \log n + \log(\log n) + \log(\log(\log n)) + \dots$$

כאשר הסיכום הוא עד ההתאפסות הראשונה (לפני שמגיעים למס' שלילי). הסדרה של לוגריתם איטרטיבי יורדת חזק מכל סדרה גיאומטרית ולכן לכל  $\varepsilon > 0$ , ועבור  $n$  מספיק גדול, מתקיים  $\log^* n = (1 + \varepsilon) \log n$ .

3. אורך ההתאמה האופייני של LZ77

נשתמש ב Kac's Lemma הטוענת, איכותית, שבכדי לקבל מאורע בעל הסתברות  $p$ , נצטרך לבצע בערך  $1/p$  ניסויים. ניתן להראות שבעיה שקולה לבעיית LZ77 היא מציאת התאמה באורך  $l$  מתוך עבר אינסופי. כמה "אחורה" בעבר נצטרך ללכת בכדי לקבל התאמה באורך  $l$  כלשהוא? עבור מקור בעל קצב אנטרופיה  $H$  כל סדרה אופיינית היא בעלת הסתברות של בערך  $2^{-Hl}$  (מתכונת ה AEP), ולכן לפי Kac's Lemma דרוש עבר של בערך  $n \approx 2^{Hl}$ . מכאן, אם נחזור לבעיית LZ77 המקורית נקבל כי עבור התאמה באורך  $l$  דרושה סדרת מקור באורך של  $l \approx \frac{\log n}{H}$  (הוכחה מדויקת נמצאת במאמר של Wyner – Ziv מ-1989).

- T. Cover, J. Thomas. "*Elements of Information Theory*". Wiley 1991.
- P. Elias . "*Universal Codeword Sets and Representations of the Integers*". IEEE Transactions on Information Theory, vol. IT-21, pp. 194 - 203, March 1975.
- A. Gersho, R. Gray . "*Vector Quantization and Signal Compression*".
- J. Gibson, T. Berger, T. Lookabaugh, D. Lindbergh, R. Baker. "*Digital Compression for Multimedia, principles and standards*" .
- A. Kolmogorov. "*Logical Basis for Information Theory and Probability Theory*" . IEEE Transactions on Information Theory, vol. IT-14 pp. 662-664, 1968.
- A. Lempel, J. Ziv. "*On the Complexity of Finite Sequences*". ". IEEE Transactions on Information Theory, vol. IT-22 pp. 75-81, 1976.
- S. Savari. "*Redundancy of the Lempel-Ziv Incremental Parsing Rule*". IEEE Transactions on Information Theory, vol. IT-43, Issue 1 pp. 9 - 21, 1997.
- B. Tunstall . "*Synthesis of noiseless compression codes*". Ph. D. Thesis, Georgia Institute of Technology, 1968.
- T. Welch. "*A technique for High Performance Data Compression*". IEEE Computer , vol. 17-6 , pp. 8 - 19, 1984.
- A. Wyner, J. Ziv. "*Some Asymptotic Properties of the Entropy of a Stationary Ergodic Data Source with Applications to Data Compression*" . IEEE Transactions on Information Theory, vol. IT-35, pp. 1250 - 1258, 1989.
- J. Ziv, A. Lempel. "*A Universal Algorithm for Data Compression*". IEEE Transactions on Information Theory, vol. IT-23, pp. 337 - 343, 1977.
- J. Ziv, A. Lempel. "*Compression of Individual Sequences via Variable Rate Coding*". IEEE Transactions on Information Theory, vol. IT-24, pp. 530 - 536, 1978.

# Lossy Compression

Data and Signal Compression  
 Dept. of Electrical Engineering - Systems, Tel-Aviv University

Summarized by Tal Philosof

## 1 Introduction

The lossy compression model is:

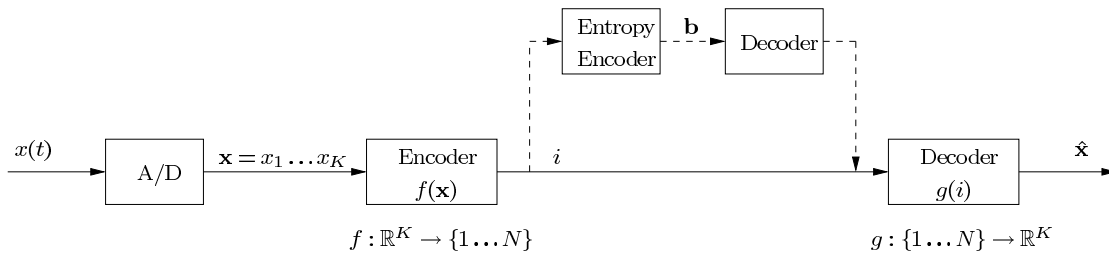


Figure 1: The lossy compression model

For  $K = 1$  scalar quantization (SQ).

For  $K > 1$  Vector quantization (VQ).

The quantization function  $\hat{\mathbf{x}} \triangleq Q(\mathbf{x}) \triangleq g(f(\mathbf{x}))$  presents the decoder and encoder operation.

$g(\cdot)$  - is one to one mapping.

$f(\cdot)$  - is many to one mapping.

The quantizer can be either fixed rate or variable rate:

$$R = \begin{cases} \frac{1}{K} \log_2 N \left[ \frac{\text{bits}}{\text{source code}} \right], & \text{fixed rate} \\ \frac{1}{K} E \{L(f(\mathbf{x}))\} \left[ \frac{\text{bits}}{\text{source code}} \right], & \text{var. rate B2V} \end{cases}$$

where  $L(f(\mathbf{x}))$  is the length of the codeword of  $f(\mathbf{x})$ .

**Distortion measures:** for a given distortion measure  $d(\mathbf{x}, \hat{\mathbf{x}})$  the distortion is

$$D = E\{d(\mathbf{X}, Q(\mathbf{X}))\}.$$

Examples of distortion measures:

1. "Per letter" -  $d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^K d(x_i, y_i)$

$d(\mathbf{x}, \mathbf{y}) = \sum_i (y_i - x_i)^2$  - squared distortion measure.

$d(\mathbf{x}, \mathbf{y}) = \sum_i (y_i - x_i)^r$  - r-th distortion measure.

2. Vectorial -

$d(\mathbf{x}, \mathbf{y}) = \|\mathbf{y} - \mathbf{x}\|^2$  - squared distortion measure.

$d(\mathbf{x}, \mathbf{y}) = (\mathbf{y} - \mathbf{x})^T W (\mathbf{y} - \mathbf{x})$  - weighted square distortion measure.

These are difference distortion measures.

### 3. Non-difference measures -

$$d(x, y) = \left[ \frac{y-x}{x} \right].$$

#### 1.1 The Quantizer Structure

$\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N$  - are the representative vectors (that is  $Q(\mathbf{x}) \in \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N\}$ ).

$S_1, S_2, \dots, S_N$  - are the quantization cells, that is  $\forall \mathbf{x} \in S_i \Rightarrow Q(\mathbf{x}) = \mathbf{y}_i$ . The cells produce a **partition** of the  $\mathbb{R}^K$  space. Therefore,

$$S_i \cap S_j = \emptyset, \forall i, j$$

$$\bigcup_{i=1}^N S_i = \mathbb{R}^K.$$

Let us consider that  $\mathbf{x}$  has a probability density function  $f(\mathbf{x})$ , thus

$$D = E \{d(Q(\mathbf{X}), \mathbf{X})\} = \int_{\mathbb{R}^K} f(\mathbf{x})d(\mathbf{x}, Q(\mathbf{x}))d\mathbf{x} = \sum_{i=1}^N \int_{S_i} f(\mathbf{x})d(\mathbf{x}, Q(\mathbf{x}))d\mathbf{x}$$

## 2 Lloyd Condition

### I. Nearest neighbor Condition -

Suppose the representative vectors  $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N$  are known, and it is needed to set the cells  $S_1, S_2, \dots, S_N$  (the decoder is known, and the encoder is needed to be designed). The optimal partition is in this case is:

$$S_i = \{\mathbf{x} : d(\mathbf{x}, \mathbf{y}_i) \leq d(\mathbf{x}, \mathbf{y}_j), \forall j\}, i = 1 \dots N \quad (1)$$

### II. Centroid Condition-

Suppose the cells  $S_1, S_2, \dots, S_N$  are known, and it is needed to set the representative vectors  $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N$  (the encoder is known, and the decoder is needed to be designed). The optimal representatives are:

$$\mathbf{y}_i = \arg \min_{\mathbf{y}} E \{d(\mathbf{X}, \mathbf{y}) | \mathbf{x} \in S_i\}, i = 1 \dots N \quad (2)$$

In case of square distortion measure the condition becomes

$$\mathbf{y}_i = E \{\mathbf{X} | \mathbf{X} \in S_i\}, \forall i$$

The centroid and nearest neighbor conditions are known as **Lloyd Conditions** (1953), which are necessary conditions.

#### 2.1 Nearest Neighbor Condition for the Euclidean Space

$$\begin{aligned} \|\mathbf{x} - \mathbf{y}_i\|^2 &\leq \|\mathbf{x} - \mathbf{y}_j\|^2 \\ \mathbf{x}^T \mathbf{x} - 2\mathbf{x}^T \mathbf{y}_i + \mathbf{y}_i^T \mathbf{y}_i &\leq \mathbf{x}^T \mathbf{x} - 2\mathbf{x}^T \mathbf{y}_j + \mathbf{y}_j^T \mathbf{y}_j \\ 2(\mathbf{y}_i - \mathbf{y}_j)^T \mathbf{x} + \|\mathbf{y}_i\|^2 - \|\mathbf{y}_j\|^2 &\leq 0. \end{aligned}$$

Let us define  $-U_{ij} \triangleq 2(\mathbf{y}_i - \mathbf{y}_j)^T \mathbf{x}$  and  $-R_{ij} \triangleq \|\mathbf{y}_i\|^2 - \|\mathbf{y}_j\|^2$ , therefore  $\mathbf{x} \in S_i$  if

$$U_{ij}^T \mathbf{x} + R_{ij} \geq 0,$$

i.e, the i-th cell is located "above" the hyperplane  $U_{ij}^T \mathbf{x} + R_{ij} = 0$ . As a consequence,  $S_i$  is a  $K$  dimensional polytope. Furthermore, the cells are convex sets, i.e,  $\forall \mathbf{x}_1, \mathbf{x}_2 \in S_i, \alpha \mathbf{x}_1 + (1 - \alpha) \mathbf{x}_2 \in S_i, 0 \leq \alpha \leq 1$ .

## 2.2 The Implication of the Nearest neighbor Condition for Square Distortion Measure

The estimation error is

$$\mathbf{e}_q = Q(\mathbf{x}) - \mathbf{x},$$

which is the quantization error. In MMSE estimation the estimation error is perpendicular to the measurements (the cell  $i$  in our case), i.e,  $\mathbf{e}_q \perp Q(\mathbf{x})$ . Therefore,

$$E\|\mathbf{X}\|^2 = E\|Q(\mathbf{X})\|^2 + E\|\mathbf{E}_q\|^2 = E\|Q(\mathbf{X})\|^2 + KD.$$

The equivalent additive noise channel model is:

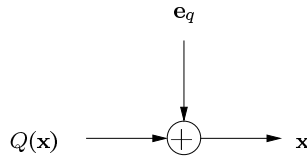


Figure 2: The equivalent additive noise channel

## 2.3 Lloyd Algorithm

The Lloyd algorithm performs (2) and (1) iteratively. Therefore, the distortion can only decrease. The algorithm converges to local minimum. If  $f(\mathbf{x})$  is log concave (i.e,  $\log(f(\mathbf{x}))$  is concave) then the Lloyd algorithm converges to the global minimum.

## 3 A Geometrical View of the Benefit of VQ Vs. SQ

For two dimensional quantizer for  $x_1, x_2$  with  $N = 6$  levels, the rate is  $R = \frac{1}{2} \log 6$  bit/sample. The equivalent quantizer after two scalar quantizations is:

$$Q(x_1, x_2) = (Q_1(x_1), Q_2(x_2)),$$

which consists of two scalar quantization. The codeword and the cells are Cartesian product of the dimension as shown in Figure 3.

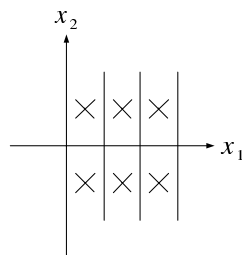


Figure 3: 2D-quantizer with cells and the codewords that are Cartesian product

**The vector quantize case:**

1. The shaping gain - the region that contains the representatives is shown in Figure 4.

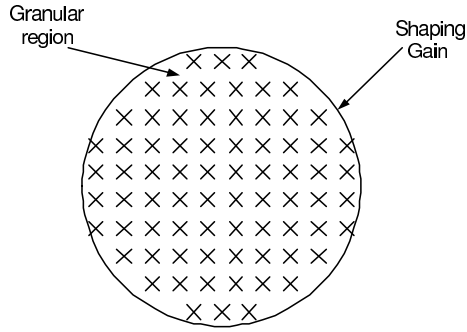


Figure 4: Shaping gain

2. The memory gain - the gain due the dependency between the dimensions is shown in Figure 5.

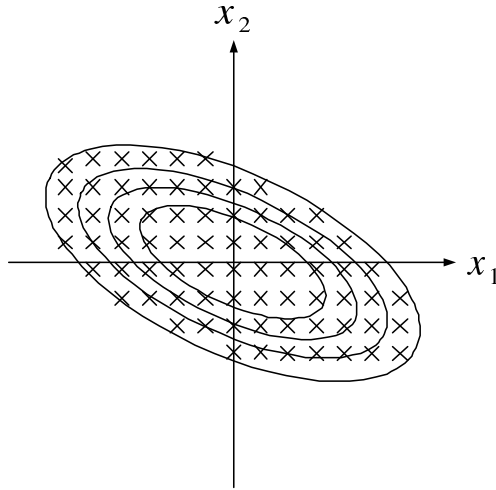


Figure 5: Memory gain

3. The granular gain - the gain due to the cells shapes is shown in Figure 6.

It should be noted that we can achieve a shaping gain even though the  $x_1, x_2$  are independent.

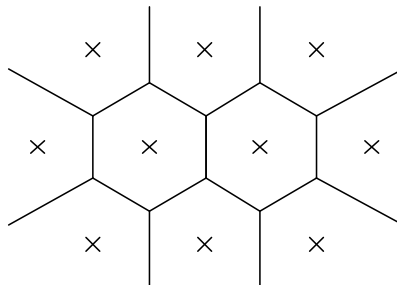


Figure 6: Granular gain

## 4 Entropy Constrained Quantization (ECQ)

The Entropy Constrained Quantization model is shown in Figure 7.

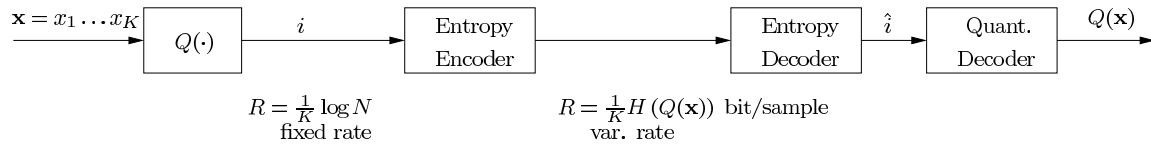


Figure 7: Entropy constrained quantization model

The problem can be written as:

$$\min_{\{N, \{y_i\}, S_i\}} D \text{ given } H \leq R,$$

using Lagrange multiplier  $\lambda$ , we get the unconstrained minimization:

$$\min_{\{N, \{y_i\}, S_i\}} D + \lambda H,$$

Graphically the distortion versus the entropy is shown in Figure 8.

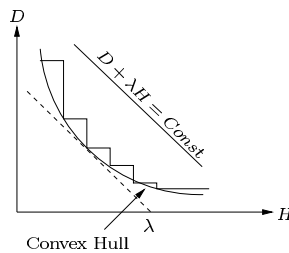


Figure 8: Entropy constrained quantization the distortion vs. entropy

Lloyd like solution is based on the “modified distortion measure” (“distortion+ $\lambda$  rate”):

$$\tilde{d}(\mathbf{x}, \mathbf{y}_i) = d(\mathbf{x}, \mathbf{y}_i) + \lambda \log \left( \frac{1}{P_i} \right),$$

where  $P_i = Pr(S_i) \triangleq \int_{\mathbf{x} \in S_i} f(\mathbf{x}) d\mathbf{x}$ , and  $d(\mathbf{x}, \mathbf{y}_i)$  is the distortion of when quantizing  $\mathbf{x}$  to  $\mathbf{y}_i$ , and  $\lambda \log \left( \frac{1}{P_i} \right)$  is the ideal codeword length of encoding the  $i$ -th cell.

For fixed  $N$  and  $\lambda \rightarrow 0 \Rightarrow$  fixed rate  $\tilde{d} = d$

For fixed  $N$  and  $\lambda \rightarrow \infty \Rightarrow$  the distortion is ignored (w.r.t the rate)

### 4.1 Iterative Algorithm

1. Calculate  $P_i, \dots, P_N$  for given cells  $S_i, \dots, S_N$ .
2. Calculate the centroid:

$$\mathbf{y}_i = \arg \min_{\mathbf{y}} E \{d(\mathbf{X}, \mathbf{y}) | \mathbf{X} \in S_i\}, \quad i = 1, \dots, N$$

3. The cells  $S_i, \dots, S_N$  are generated using the nearest neighbor condition using the “distance measure”  $\tilde{d}(\cdot, \cdot)$  with respect to  $\mathbf{y}_i, i = 1, \dots, N$  (The  $P_i, i = 1 \dots N$  are from phase 1).
4. return to phase 1.

The algorithm is sub-optimal, however the algorithm is improving in each step (phase 3 improves  $D + \lambda R$ ), therefore it will converge to local minimum.

## 5 Uniform Scalar Quantization

Uniform scalar quantization is shown in Figure 9.

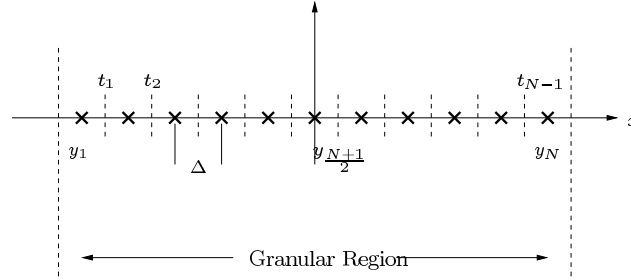


Figure 9: Scalar uniform quantizer

$\Delta$  - is the quantizer step.

$N$  - number of levels.

$R$  - is  $\log_2 N$  bit/sample.

The quantizer (mid-thread) function is:

$$Q(x) = \begin{cases} i & i\Delta - \frac{\Delta}{2} \leq x \leq i\Delta + \frac{\Delta}{2} \text{ (gran region)} \\ \text{edge point} & \text{overload (ol) region} \end{cases}$$

Unlike in Lloyd, the quantizer is robust, i.e, does not depend on the distribution (ignoring the fact that  $N \frac{\Delta}{2} = \gamma \sigma$ ).

The distortion is

$$D = E \{d(Q(X), X) | X \in \text{granular}\} Pr(\text{granular}) + E \{d(Q(X), X) | X \in \text{ol}\} Pr(\text{ol}).$$

### 5.1 High Resolution Quantization

In high resolution quantization (HRQ), we assume that:

1. In the granular region the source is distributed “almost” uniformly in each cell, i.e,  $\frac{\Delta f}{f} \ll 1$  ( $\Delta f$  is the tilt of  $f$  in the cell).
2. The distortion in overload region is negligible with respect to the distortion in the granular region, thus for difference distortion measure

$$D \simeq D_{\text{grain}} \simeq \frac{1}{\Delta} \int_{\Delta/2}^{\Delta/2} d(\alpha) d\alpha = \begin{cases} \frac{\Delta^2}{12} & ()^2 \text{ measure} \\ K_r \Delta^r & ()^r \text{ measure} \end{cases}$$

### 5.2 Non-Uniform Scalar Quantization

The following scheme performs non-uniform scalar quantization (usually used for uniform quantization in logarithmic scale (A-law,  $\mu$ -law)):



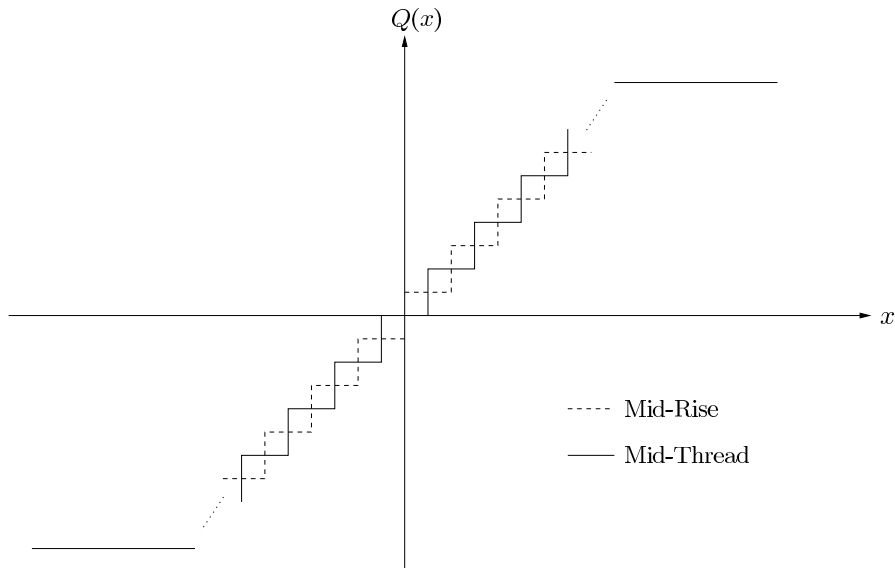


Figure 10: Mid-rise and Mid-thread quantizers

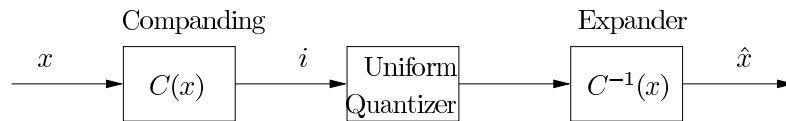


Figure 11: Non-uniform Quantizer

## 6 Uniform Vector Quantization

In uniform vector quantization we need to find **space filling polytopes** which are **tessellating** the space without overlapping between polytopes. Each polytope is a quantizer cell and under some conditions the representatives of the space filling polytopes form “linear” set, in which case the set is called **lattice**.

### 6.1 Lattice Definitions and Properties

1. **Lattice** - A set of discrete points in  $\mathbb{R}^K$  is a lattice  $\Lambda$ , if  $\mathbf{y}_i + \mathbf{y}_j \in \Lambda$ , for all  $\mathbf{y}_i, \mathbf{y}_j \in \Lambda$ .
2. **Generator matrix** -  $G$  is a generator matrix of the lattice  $\Lambda$ , then

$$G = (\mathbf{g}_1 \ \mathbf{g}_2 \ \dots \ \mathbf{g}_K).$$

and the lattice is defined as

$$\Lambda = \{\mathbf{x} : \mathbf{x} = G\mathbf{m}\},$$

where  $\mathbf{m} = (m_1 \dots m_K)$ ,  $m_i \in \{0, \pm 1, \pm 2, \dots\}$ , thus  $\Lambda$  is all the integer linear combinations of  $\mathbf{g}_1, \dots, \mathbf{g}_K$ . The vectors  $\mathbf{g}_1, \dots, \mathbf{g}_K$  are called basis vectors. The basis vectors are not unique; also  $G' = GB$  is a basis where  $B$  is an integer matrix

$$B = \begin{pmatrix} 1 & -1 \\ 0 & 1 \end{pmatrix}$$

with  $\det(B) = 1$ .

3. **Partition** - Any lattice code is also attached to partition. The partition associated with a given lattice is not unique. Partition of nearest neighbor for Euclidean distance is known as Voronoi partition.

4. **Normalized second moment ( $\Lambda$ , partition)** - Let us define the lattice basic cell  $\mathcal{V}$  where  $\mathcal{V}\{\mathbf{x} :$

$Q(\mathbf{x}) = \mathbf{0}$ . For any  $K$  dimensional set  $S$ , the normalized second moment is define as:

$$G(S) = \frac{1}{K} \left\{ \frac{\text{second moment for uniform pdf over } S}{(\text{volume } S)^{2/K}} \right\}.$$

The normalized second moment of lattice is defined as

$$G(\Lambda) = G(\mathcal{V}).$$

For  $V$  which is the volume of the basic cell, i.e,

$$V = \text{Vol}(\mathcal{V}) = \det(G) = \int_{\mathbf{x} \in \mathcal{V}} d\mathbf{x},$$

where  $G$  is the lattice generator matrix. The normalized second moment of the lattice is

$$G(\Lambda) \triangleq \frac{\frac{1}{K} \int_{\mathbf{x} \in \mathcal{V}} \|\mathbf{x}\|^2 d\mathbf{x}}{V^{1+2/K}}.$$

For uniform source and square distortion measure and lattice quantizer, the distortion is

$$D = G(\Lambda)V^{2/K}.$$

**5. Isoperimetric Inequality** - Sphere has the minimal second moment from all the shapes with equal volume ( $V$ ), i.e,

$$\frac{1}{V} \int_S \|\mathbf{x}\|^2 d\mathbf{x} \geq \frac{1}{V} \int_{\mathcal{B}} \|\mathbf{x}\|^2 d\mathbf{x},$$

where  $\mathcal{B}$  is a sphere with volume  $V$ . The inequality can be written with respect to normalized second moment

$$G(\Lambda) \geq G_K^* = G(\mathcal{B})$$

Some known normalized second moment for spheres:

$$\begin{aligned} K = 1 & : G = \frac{1}{\Delta} = \frac{1}{12} \\ K = 2 & : G = \frac{1}{4\pi} \\ K \rightarrow \infty & : G = \frac{1}{2\pi e} \approx \frac{1}{17}. \end{aligned}$$

## 7 Sphere Packing / Covering and Lattices

### 7.1 Covering and Packing using Spheres

**Covering** - Each point in the space is covered at least by one sphere.

An efficient covering  $\Leftrightarrow$  minimum spheres.

An efficient covering  $\Leftrightarrow$  minimum overlapping.

**Packing** - There is no overlap between spheres (the spheres are not flexible).

An efficient packing  $\Leftrightarrow$  maximum spheres.

An efficient packing  $\Leftrightarrow$  minimum wasted volume.

Let us define the following values:

$$\begin{aligned} \Theta = \text{covering thickness} &= \frac{\text{spheres volume}}{\text{covered volume}} = \frac{N \cdot \text{sphere volume}}{\text{covered volume}} \\ \Delta = \text{packing density} &= \frac{\text{spheres volume}}{\text{packing volume space}} \end{aligned}$$

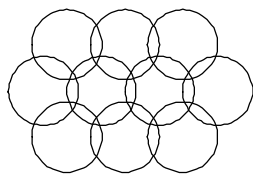


Figure 12: Covering using sphere

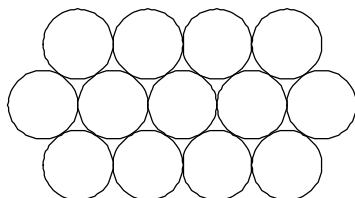


Figure 13: Packing using sphere

## 7.2 Lattices

In case that the spheres are laid on the lattice points the covering thickness and the packing density is defined as :

$$\Theta = \frac{\text{volume of circumscribed sphere of Voronoi cell}}{\text{Voronoi cell volume}}$$

$$\Delta = \frac{\text{volume of inscribed sphere in Voronoi cell}}{\text{Voronoi cell volume}}$$

For large dimension the covering problem is equivalent to the quantization problem with MSE distortion measure.

The packing problem is similar with the modulation problem (constellation points in signal space) for AWGN channel. However, packing is not equivalent to modulation even in large dimension; it more pessimistic than Shannon theory.

For  $K \rightarrow \infty$ :

$$\frac{1}{K} \log \Theta_K \rightarrow 0$$

$$\frac{1}{K} \log \Delta_K \rightarrow \log 0.66 = -0.599.$$

The covering thickness is increased sub exponential, thus it is good also for quantization. The packing density is decreased sub exponential, however for AWGN channel coding we would expected it will converge to  $\log(\frac{1}{2})$ .

## שתי תורות אסימפטוטיות של קוונטיזציה: במימד וברזולוציה

סיכום נערך ע"י אלי חיים

### אסימפטוטיות במימד – שנון

יהי  $\mathbf{X} = (X_1, X_2, \dots, X_k)$  וקטור אקראי מפולג רציף.

תכונת ה-Asymptotic Equipartition Property (AEP) למקור ארגודי עם פילוג רציף:

$$A_k^{(\varepsilon)} \triangleq \left\{ \mathbf{x} \mid 2^{-k\left[\frac{1}{k}h(\mathbf{X})+\varepsilon\right]} \leq f(\mathbf{x}) \leq 2^{-k\left[\frac{1}{k}h(\mathbf{X})-\varepsilon\right]} \right\}$$

$$\Pr(\mathbf{X} \in A_k^{(\varepsilon)}) \xrightarrow[k \rightarrow \infty]{} 1$$

$$\text{Vol}(A_k^{(\varepsilon)}) \cong 2^{h(\mathbf{X})}$$

כלומר עבור  $k$  גדול, המקור מתפלג בקירוב אחיד על  $A_k^{(\varepsilon)}$ . לכן הקוונטיזציה של כל מקור ארגודי הופכת להיות קוונטיזציה של מקור אחיד (בתחום רב-מימדי).

### דוגמא - מקור גאואסי לבן

$$\mathbf{X} \sim N(\mathbf{0}, \sigma^2 I_k)$$

$$f(\mathbf{x}) \propto e^{-\frac{\|\mathbf{x}\|^2}{2\sigma^2}}$$

מהחוק החלש של המספרים הגדולים (LLN):

$$\frac{1}{k} \|\mathbf{X}\|^2 = \frac{1}{k} \sum_{i=1}^k X_i^2 \xrightarrow[k \rightarrow \infty]{P} \sigma^2$$

ולכן בהסתברות גבוהה (עבור  $k$  מספיק גדול):

$$\|\mathbf{X}\|^2 = k\sigma^2$$

כלומר בהסתברות גבוהה נקבל כדור  $k$ -מימדי בעל רדיוס  $\sqrt{k\sigma^2}$ .

מאיזוטרופיות הפילוג הגאואסי נובע ש- $\mathbf{X}$  מתפלג אחיד על פני מעטפת כדור ברדיוס  $k\sigma^2$ . במימד  $k$  גבוה זה שקול לפילוג אחיד על פני הכדור כולו, מכיוון שיחס הנפחים של כדור  $k$ -מימדי ברדיוס

$r - \varepsilon$  ושל כדור  $k$ -מימדי ברדיוס  $r$  שואף ל-0 כאשר  $k \rightarrow \infty$ :

$$\frac{\text{Vol}(\text{ball}_k(r-\varepsilon))}{\text{Vol}(\text{ball}_k(r))} \xrightarrow[k \rightarrow \infty]{} 0$$

נניח כי תרומת ה"זנב" לעיוות MSE היא זניחה:

$$D_{\text{overload}} = \Pr(\mathbf{X} \notin A_k^{(\varepsilon)}) \cdot \frac{1}{k} E \left\{ \|\mathbf{X}\|^2 \mid \mathbf{X} \notin A_k^{(\varepsilon)} \right\} \xrightarrow[k \rightarrow \infty]{} 0$$

מהנחה זו ומתכונת ה-AEP נובע שקוונטיזציה של  $\mathbf{X}$  שקולה לקוונטיזציה של הקבוצה  $A_k^{(\varepsilon)}$  ביחס לפילוג אחיד.

בנה קוונטיזציה בעל אזור גרעיני  $A_k^{(\varepsilon)}$ , בעל  $N$  נקודות קוונטיזציה וחלוקה כך ש:

$$S_i \triangleq i^{\text{th}} \text{ cell} \cap A_k^{(\varepsilon)}$$

$$V_i \triangleq \text{Volume}(S_i)$$

נחשב את העיוות שנוצר:

$$D_{granular} = \frac{1}{k} E \left\{ \|\mathbf{X} - Q(\mathbf{X})\|^2 \mid \text{no overload} \right\} = \sum_{i=1}^N p_i D_i$$

$$p_i \triangleq \Pr(\mathbf{X} \in S_i) = \int_{S_i} f(\mathbf{x}) d\mathbf{x} \underset{\text{AEP}}{\approx} \frac{V_i}{V_{granular}}$$

$$D_i \triangleq \frac{1}{k} E \left\{ \|\mathbf{X} - \mathbf{y}_i\|^2 \mid \mathbf{X} \in S_i \right\} =$$

$$= \frac{1}{k} \int_{S_i} \|\mathbf{x} - \mathbf{y}_i\|^2 f(\mathbf{x}) d\mathbf{x} \cong$$

$$\underset{\text{AEP}}{\approx} \frac{1}{k} \frac{1}{V_i} \int_{S_i} \|\mathbf{x} - \mathbf{y}_i\|^2 d\mathbf{x} \geq$$

$$\underset{\text{isoperimetric inequality}}{\geq} G_k^* V_i^{2/k}$$

isoperimetric inequality

לפי אי-שוויון איזופרימטרי, העיוות הממוצע לתא הינו מינימלי עבור תא כדורי, ובמקרה זה:

$$D_i = G_k^* V_i^{2/k}$$

כאשר:

$$G_k^* \triangleq \frac{1}{k} \frac{\int_B \|\mathbf{x}\|^2 d\mathbf{x}}{V^{2/k}}$$

ו-B הוא כדור k-מימדי בעל נפח V.

אזי:

$$D_{granular} = \sum_{i=1}^N p_i D_i = \sum_{i=1}^N \frac{V_i}{V_{granular}} G_k^* V_i^{2/k} = \frac{G_k^*}{V_{granular}} \sum_{i=1}^N V_i^{1+2/k} =$$

$$= \frac{G_k^*}{V_{granular}} N \left( \frac{1}{N} \sum_{i=1}^N V_i^{1+2/k} \right) \underset{\text{Jensen}}{\geq} \frac{G_k^*}{V_{granular}} N \left( \frac{1}{N} \sum_{i=1}^N V_i \right)^{1+2/k}$$

$$= \frac{G_k^*}{V_{granular}} N \left( \frac{V_{granular}}{N} \right)^{1+2/k} =$$

$$= G_k^* \left( \frac{V_{granular}}{N} \right)^{2/k}$$

ושוויון אמ"מ:

$$V_i = \frac{V_{granular}}{N}$$

כלומר אם נחלק את האזור הגרעיני לתאים בצורת כדורים בעלי נפחים זהים, אזי נקבל קוונטיזר בעל ביצועים אופטימליים.

עבור מקור חסר זכרון:

$$V_{granular} = Vol(A_k^{(\epsilon)}) \cong 2^{h(X)} \underset{\substack{\uparrow \\ \text{memoryless source}}}{=} 2^{kh(X)}$$

$$N = 2^{kR}$$

$$\begin{aligned} D_{granular} &\geq G_k^* \left( \frac{V_{granular}}{N} \right)^{2/k} \cong G_k^* (2^{k[h(X)-R]})^{2/k} = \\ &= G_k^* 2^{2[h(X)-R]} \geq \frac{1}{2\pi e} 2^{2[h(X)-R]} \end{aligned}$$

ועבור  $k \rightarrow \infty$  אי-השוויון האחרון הופך לשוויון. במקרה זה:

$$R = h(X) - \frac{1}{2} \log(2\pi e D_{granular})$$

**החסם התחתון של שנון (SLB) Shannon Lower Bound** עבור מדד MSE:

$$R(D) \geq h(X) - \frac{1}{2} \log(2\pi e D)$$

**מסקנה:** בכדי להשיג את ה-SLB צריך קוונטיזר שהתאים שלו בצורתם כדורים ושווי נפח ( $k \rightarrow \infty$ ).

**הערות:**

1. לכל מדד עיוות הפרשי  $d(\mathbf{x}, \hat{\mathbf{x}}) = \rho(\hat{\mathbf{x}} - \mathbf{x})$  ניתן להגדיר כדורים מוכללים.
2. היינו מצפים לקבל תאים בצורת מצולעים (תאי Voronoi), אך עבור  $k \rightarrow \infty$  הם שואפים להיות קרובים יותר ויותר לכדורים.

## אסימפטוטיות ברזולוציה – קוונטיזציה ברזולוציה גבוהה (High Resolution Quantization)

בקוונטיזציה ברזולוציה גבוהה מניחים שהתאים מספיק צפופים כך שבכל תא פונקציית צפיפות הפילוג של המקור היא בקירוב אחידה.

הקוונטיזר מוגדר ע"י פונקציית צפיפות נקודות קוונטיזציה ליחידת נפח מנורמלת:

$$\begin{aligned} \lambda : \mathbb{R}^k &\rightarrow \mathbb{R}_+ \\ \lambda(\mathbf{x}) &\geq 0 \\ \int_{\mathbb{R}^k} \lambda(\mathbf{x}) d\mathbf{x} &= 1 \end{aligned}$$

נסמן ב-N את מספר נקודות השחזור הכולל.

מספר נקודות השחזור באזור S במרחב נתון ע"י:

$$N(S) \cong N \int_S \lambda(\mathbf{x}) d\mathbf{x}$$

בסביבת x, צפיפות הנקודות ליחידת שטח הינה  $N\lambda(\mathbf{x})$ .

הנפח של תאי הקוונטיזציה בסביבת x הוא בקירוב  $\frac{1}{N\lambda(\mathbf{x})}$ .

**במקרה הסקלרי**  $\lambda(x)$  מספיקה להגדרת קוונטיזר, מכיוון שהתאים הינם אינטרוולים (בגודל  $\frac{1}{N\lambda(x)}$ , ונקודות השחזור  $y_i$  הינן במרכז התא).

**במקרה הוקטורי** – צריך בנוסף לנפח התאים להגדיר את צורתם. נניח כי  $\lambda(\mathbf{x})$  מאפנן את צפיפות הנקודות של קוונטיזר וקטורי אחיד (ריצוף, שריג).

**המטרה:** לכתוב את  $D$  (ואת  $R$ ) כפונקציה של  $\lambda(\cdot)$  ו- $f(\cdot)$  ולמצוא:  $\lambda(\cdot)$  אופטימלי עבור קוונטיזציה בקצב קבוע, ו- $\lambda(\cdot)$  אופטימלי עבור קוונטיזציה עם אילוץ אנטרופיה.

$$f(\mathbf{x}|\mathbf{x} \in S_i) \approx \text{Uniform}(S_i)$$

$$V_i \triangleq \text{Vol}(S_i)$$

$$p_i \triangleq \Pr(\mathbf{X} \in S_i) \cong V_i f(y_i) \cong \frac{f(y_i)}{N\lambda(y_i)}$$

$$D_i \triangleq \frac{1}{k} E\{d(\mathbf{X}, Q(\mathbf{X}))|\mathbf{X} \in S_i\} \cong \underset{G(S)=G}{V_i^{2/k} G} \cong G \left[ \frac{1}{N\lambda(y_i)} \right]^{2/k}$$

כאשר הנחנו שהקוונטיזר הינו "אפנון" של ריצוף עם תא  $S: G(S)=G$ .

### קוונטיזר בקצב קבוע – Fixed Rate

$$R = \frac{1}{k} \log N$$

$$D = \sum_{i=1}^N p_i D_i \cong \sum_{i=1}^N V_i f(y_i) G \left[ \frac{1}{N\lambda(y_i)} \right]^{2/k} \cong \underset{\text{Riemann}}{G} \int_{\mathbb{R}^k} \frac{f(\mathbf{x})}{[N\lambda(\mathbf{x})]^{2/k}} d\mathbf{x}$$

כיוון ש:  $y_i$  דוגם את  $f(\mathbf{x}) \left[ \frac{1}{\lambda(\mathbf{x})} \right]^{2/k}$  עם תא בגודל  $V_i$ , המעבר מתקבל עפ"י אינטגרציית רימן. ניתן למצוא  $\lambda(\cdot)$  אופטימלית אשר מביאה את  $D$  למינימום ע"י:

1. חשבון וריאציות (גזירה ביחס ל- $\lambda(\cdot)$ ).

2. אי-שוויון הולדר: עבור  $p, q > 1$  אשר מקיימים  $\frac{1}{p} + \frac{1}{q} = 1$  מתקיים  $(\|f\|_p \triangleq [\int |f(x)|^p dx]^{1/p})$

$$\|fg\|_1 \leq \|f\|_p \|g\|_q \quad \text{ושוויון אמ"מ: } |g(x)| = c|f(x)|^{p-1}$$

לפי אי-שוויון הולדר:

$$D \geq D_{MSE}^{\text{opt}} = \frac{G}{N^{2/k}} \left[ \int_{\mathbb{R}^k} f(\mathbf{x})^{\frac{k}{k+2}} d\mathbf{x} \right]^{\frac{k+2}{k}} = \frac{G}{N^{2/k}} \|f\|_{\frac{k}{k+2}}$$

ושוויון אמ"מ:

$$\lambda(\mathbf{x}) \propto f(\mathbf{x})^{\frac{k}{k+2}}$$

עבור  $k=1$  הפתרון של Bennett (1948):

$$D^{\text{opt}} = \frac{G}{N^2} \|f\|_{\frac{1}{3}}; \quad \lambda(\mathbf{x}) \propto f(\mathbf{x})^{\frac{1}{3}}$$

### הערות:

1. התאים המתקבלים אינם שווים בנפחם, אך תרומת העיוות שלהם שווה.
2. אם היינו בנוסף לוקחים  $k \rightarrow \infty$ , היינו מקבלים  $\lambda(\cdot)$  אחיד (AEP).

## קוונטיזציה בקצב משתנה (עם אילוץ אנטרופיה) – Variable Rate (Entropy Constrained Quantization)

$$\begin{aligned}
 \overline{\text{Rate}} = H(Q) &= -\sum_{i=1}^N p_i \log p_i \cong \\
 &\cong \int_{\mathbb{R}^k} f(\mathbf{x}) \log \frac{1}{N\lambda(\mathbf{x})} d\mathbf{x} = \\
 &\quad \uparrow \\
 &\quad p_i \cong f(\mathbf{y}_i) V_i \cong \frac{f(\mathbf{y}_i)}{N\lambda(\mathbf{x})} \\
 &= h(\mathbf{x}) - \frac{k}{2} \int_{\mathbb{R}^k} f(\mathbf{x}) \log \frac{1}{[N\lambda(\mathbf{x})]^{2k}} d\mathbf{x} = \\
 &\quad \uparrow \\
 &\quad \geq h(\mathbf{x}) - \frac{k}{2} \log \left\{ \int_{\mathbb{R}^k} f(\mathbf{x}) \frac{1}{[N\lambda(\mathbf{x})]^{2k}} d\mathbf{x} \right\} = \\
 &\quad \text{Jensen} \\
 &= h(\mathbf{x}) - \frac{k}{2} \log \frac{D}{G}
 \end{aligned}$$

וזהו חסם תחתון עבור קוונטיזר עם תאים לפי  $\lambda(\cdot)$ . שוויון מושג אמ"מ הפונקציה שמבצעים עליה תוחלת הינה קבועה, כלומר אמ"מ  $\lambda(\cdot) = \text{const}$ .

### הערות:

1. תכונת  $\int_{\mathbb{R}^k} \lambda(\mathbf{x}) d\mathbf{x} = 1$  מושגת עבור  $\lambda(\cdot) = \text{const}$  באזור הגרעיני.
2. קיבלנו קוונטיזר אחיד. מקודד האנטרופיה מבצע את העבודה.
3. הקוונטיזר שהתקבל הינו רובאסטי, כלומר אינו תלוי בפילוג המקור. שריג (או ריצוף) טוב יהיה טוב לכל מקור. קידוד האנטרופיה אופטימלי לפילוג המושרה.
4. ניתן להראות כי העיוות בקוונטיזציה בקצב משתנה נמוך יותר מאשר בקוונטיזציה בקצב קבוע.
  - i. אפילו קידוד אנטרופיה לקוונטיזר האופטימלי בקצב קבוע ישפר קצב.
  - ii. בנוסף, הקוונטיזר האופטימלי בקצב משתנה הוא שונה מאשר בקצב קבוע.

### רווח העלייה במימד

#### קוונטיזציה סקלרית ( $k=1$ )

3. קצב קבוע:  $N = 2^R$

$d(x; \hat{x}) = \ \hat{x} - x\ ^2$	$\hat{x} - x$	$\ \hat{x} - x\ ^r$
$\lambda(x) \propto$	$f(x)^{\frac{1}{3}}$	$f(x)^{\frac{1}{1+r}}$
$D^{\text{opt}} =$	$\frac{1}{12N^2} \ f\ _{\frac{1}{3}}$	$\frac{\gamma(r)}{N^r} \ f\ _{\frac{1}{1+r}}$

4. קצב משתנה:  $R = H(Q(X))$



$$\lambda^{\text{opt}}(x) = \text{Uniform}(-\infty, +\infty)$$

$$D_2^{\text{opt}} = \frac{1}{12} 2^{2h(X)} 2^{-2R}$$

$$D_r^{\text{opt}} = \gamma(r) 2^{rh(X)} 2^{-rR}$$

### קוונטיזציה וקטורית (k-מימדית)

1. קצב קבוע:  $N = 2^{kR}$

$d(\mathbf{x}; \hat{\mathbf{x}}) = \ \hat{\mathbf{x}} - \mathbf{x}\ ^2$	$\ \hat{\mathbf{x}} - \mathbf{x}\ ^2$	$\ \hat{\mathbf{x}} - \mathbf{x}\ ^r$
$\lambda(\mathbf{x}) \propto f(\mathbf{x})^{\frac{k}{k+2}}$	$f(\mathbf{x})^{\frac{k}{k+2}}$	$f(\mathbf{x})^{\frac{k}{k+r}}$
$D^{\text{opt}} = \frac{G(\Lambda)}{N^{2/k}} \ f\ _{\frac{k}{k+2}}$	$\frac{G(\Lambda)}{N^{2/k}} \ f\ _{\frac{k}{k+2}}$	$\gamma(\Lambda, r, \nu) \ f\ _{\frac{k}{k+r}} 2^{-rR}$

כאשר D הוא עיוות למימד:

$$D = \frac{1}{k} E \|\hat{\mathbf{X}} - \mathbf{X}\|_v^r$$

2. קצב משתנה:  $R = \frac{1}{k} H(Q(\mathbf{X}))$

$$\lambda^{\text{opt}}(\mathbf{x}) = \text{Uniform}(-\infty, +\infty)$$

$$D_2^{\text{opt}} = G(\Lambda) 2^{2\frac{1}{k}h(\mathbf{X})} 2^{-2R}$$

$$D_r^{\text{opt}} = \gamma(\Lambda, r, \nu) 2^{r\frac{1}{k}h(\mathbf{X})} 2^{-rR}$$

### חסם שאנון ( $D \rightarrow 0, k \rightarrow \infty$ )

$$R(D) = R_{\text{SLB}}(D) = h(X) - h_{\text{max}}(D)$$

כאשר  $h_{\text{max}}(D)$  - maximum entropy:

$$h_{\text{max}}(D) \triangleq \max_{\{Z: E\rho(Z) \leq D\}} h(Z)$$

עבור  $\rho(z) = z^2$ :

$$h_{\text{max}}(D) = h(N(0, D)) = \frac{1}{2} \log(2\pi e D)$$

עבור MSE:

$$D_{\text{Shannon}}^{\text{opt}} = \underbrace{\frac{2^{2\bar{h}(x^\infty)}}{2\pi e}}_{\text{entropy-power}} 2^{-2R}$$

ועבור r כללי:

$$D_{\text{Shannon}}^{\text{opt}} = \gamma(\infty, r, r) 2^{r\bar{h}(x^\infty)} 2^{-rR}$$

דוגמא – מקור גאוסני לבן:

$$X \sim N(0, \sigma^2)$$

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

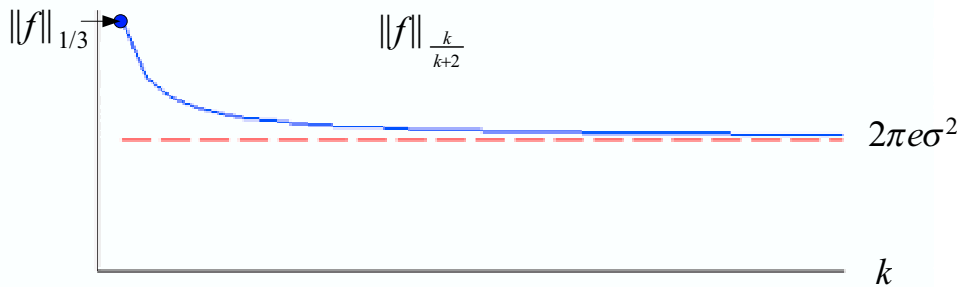
$$f(\mathbf{x}) = \prod_{i=1}^k f(x_i)$$

$$\|f\|_{\frac{1}{3}} = 2\pi 3^{1.5} \sigma^2$$

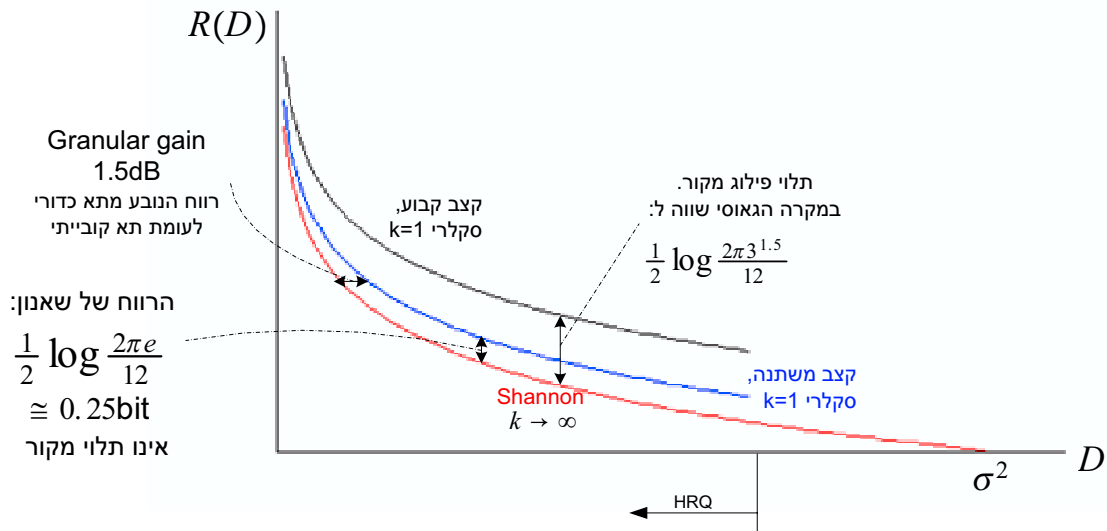
$$h(x) = \frac{1}{2} \log(2\pi e \sigma^2)$$

$$2^{2h(x)} = 2\pi e \sigma^2$$

תלות נורמת הפילוג במימד:



ביצועי קצב-עיוות:



## Shannon Lower Bound

$$R(D) = \min_{\{\hat{X}|E\rho(X-\hat{X})\leq D\}} I(X;\hat{X})$$

$$I(X;\hat{X}) = h(X) - h(X|\hat{X}) = h(X) - h(X - \hat{X}|X) \geq h(X) - h(X - \hat{X})$$

$$h_{\max}(D) \triangleq \max_{\{Z:E\rho(Z)\leq D\}} h(Z)$$

בהנחה ש- $h(x)$  קיים. לכן:

$$R(D) \geq h(X) - h_{\max}(D) \triangleq R_{\text{SLB}}(D)$$

זהו השקול במימד גבוה (צורת שאנון) של חסם הכדורים (גאוסיאן ~ כדור, אנטרופיה מקסימלית ~ כדור תחת נורמת  $\rho$ ).

הדיקות ה-S.L.B.: לכל מקור עם  $h(X)$  סופי:

$$\left[ R(D) - R_{\text{SLB}}(D) \right] \xrightarrow{D \rightarrow 0} 0$$

עבור ערוץ רעש אדיטיבי  $X = \hat{X} + Z^*$  מתקבל שוויון בחסם. עבור  $D$  כללי לא תמיד ניתן לקיים קשר אדיטיבי זה, אך בגבול  $D \rightarrow 0$  זה תמיד אפשרי.

## היתרון הווקטורי בקוונטיזציה (Lookabaugh & Gray 1989)

נניח מקור סטציונרי (במובן הצר)  $X_1, X_2, \dots, X_k, \dots$  עם פילוג שולי  $\tilde{f}(x)$ .

נגדיר מקור חסר זכרון המושרה מ- $\tilde{f}(x)$ . הפילוג המתקבל הוא:

$$f^*(\mathbf{x}) = \prod_{i=1}^k \tilde{f}(x_i)$$

עבור קצב קבוע:

$$\frac{D_{F.R.}^{(k=1)}}{D_{F.R.}^{(k)}} = \frac{1}{12G_k} \frac{\|\tilde{f}\|_{1/3}}{\|\tilde{f}\|_{\frac{k}{k+2}}} = \underbrace{\frac{1}{12G_k}}_{\text{space-filling gain}} \cdot \underbrace{\frac{\|\tilde{f}\|_{1/3}}{\|\tilde{f}\|_{\frac{k}{k+2}}}}_{\text{shaping gain}} \cdot \underbrace{\frac{\|f^*\|_{\frac{k}{k+2}}}{\|\tilde{f}\|_{\frac{k}{k+2}}}}_{\text{memory gain}}$$

- Space-filling gain – רווח גרנולרי: הרווח ממילוי המרחב
- Shaping gain: הרווח מתפירה של התחום הגרעיני לתחום האופייני (AEP).
- Memory gain – רווח הזכרון: הרווח מכך שלמקור יש תלות בין הרכיבים

עבור קצב משתנה:

$$\frac{D_{V.R.}^{(k=1)}}{D_{V.R.}^{(k)}} = \underbrace{\frac{1}{12G_k}}_{\text{space-filling gain}} \cdot \underbrace{2^{2[h(X_1) - \frac{1}{k}h(X_1, \dots, X_k)]}}_{\text{memory gain}}$$

בקידוד אנטרופיה לא קיים ה-shaping gain.

## **References**

- Gersho “*Asymptotically Optimal Block Quantization*”, IEEE Transactions on Information Theory, July 1979
- Gray & Neuhoff: “*Quantization*”, IEEE Transactions on Information Theory, October 1998

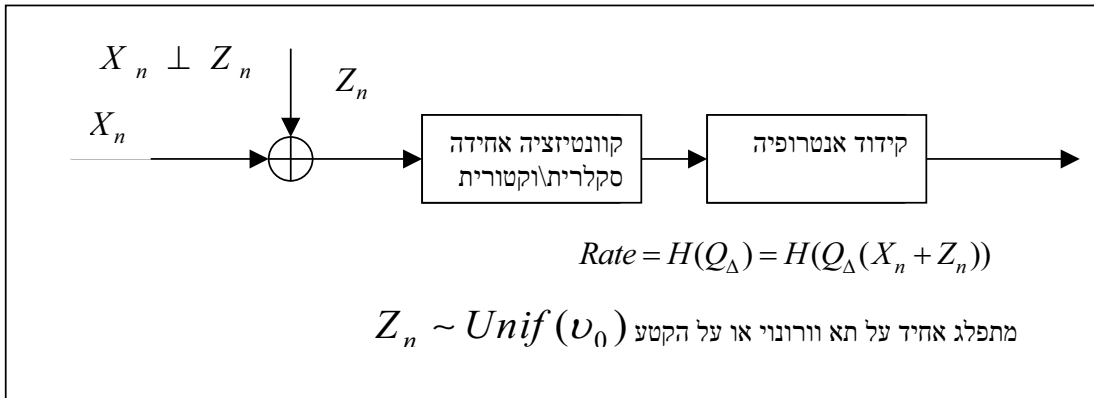
# Entropy Coded Dithered Lattice Quantizer

Summary of Prof R. Zamir class, by T. Be'ery

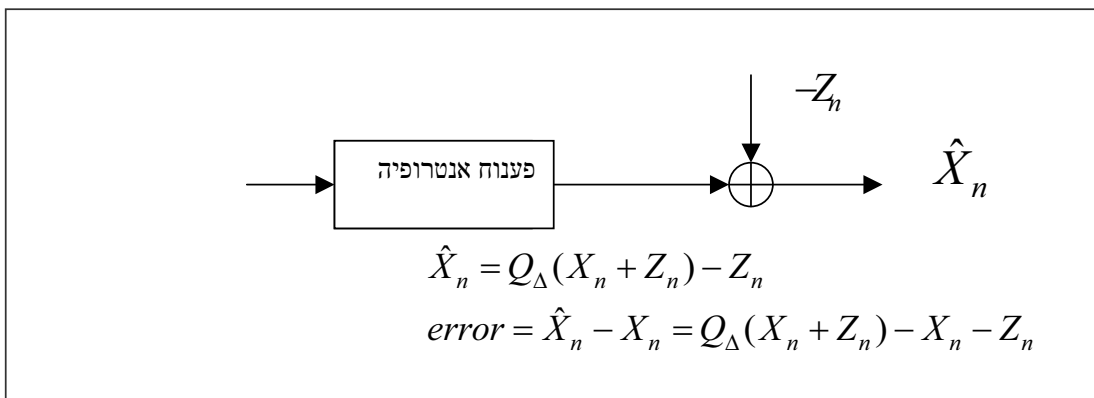
המטרה – התנהגות HRQ בכל רזולוציה.

Dither – רעש ידוע למקודד ולמפענח.

מקודד:



מפענח:



טענות:

1. שגיאת הקוונטיזציה של ה EDCQ היא בת"ס במקור ומתפלגת אחיד על פני תא הקוונטיזציה

$$D = E(\rho(Z)) \Leftarrow (\text{universal quantizer}) \Leftarrow \left(-\frac{\Delta}{2}, \frac{\Delta}{2}\right)$$

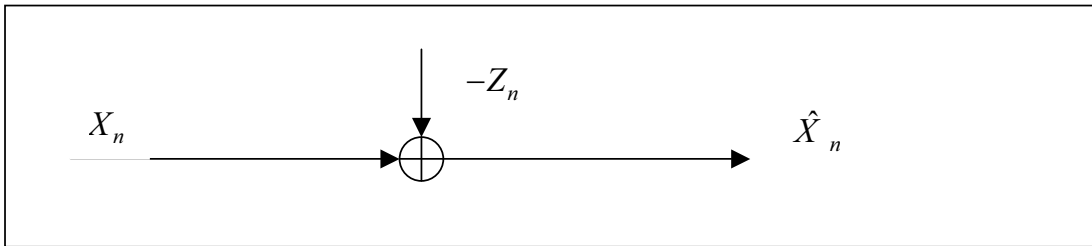
$$\text{error} = \hat{X}_n - X_n = Q_\Delta(X_n + Z_n) - X_n - Z_n \text{ : (נתבונן על השגיאה)}$$

מכיוון שהמשרעת של  $Z_n$  היא בגודל תא הקוונטיזציה  $\left(-\frac{\Delta}{2}, \frac{\Delta}{2}\right)$ ,  $Q_\Delta(X_n + Z_n) \approx X_n$ ,

וכמוכן  $Z_n$  מתפלג אחיד, לכן כל השגיאה מתפלגת באופן אחיד. (

$$2. \text{Rate} \triangleq H(Q|Z) = I(X_n; X_n - Z_n)$$

דומה לבעיית ה  $R(D)$ .



$$R(D) = \min_{\{\hat{x}: E\rho(\hat{x}-x) \leq D\}} I(x; \hat{x}) \leq I(x; x - z)$$

תוצאות על ההפסד של ה EDCQ:

1. ברזולוציה גבוהה:

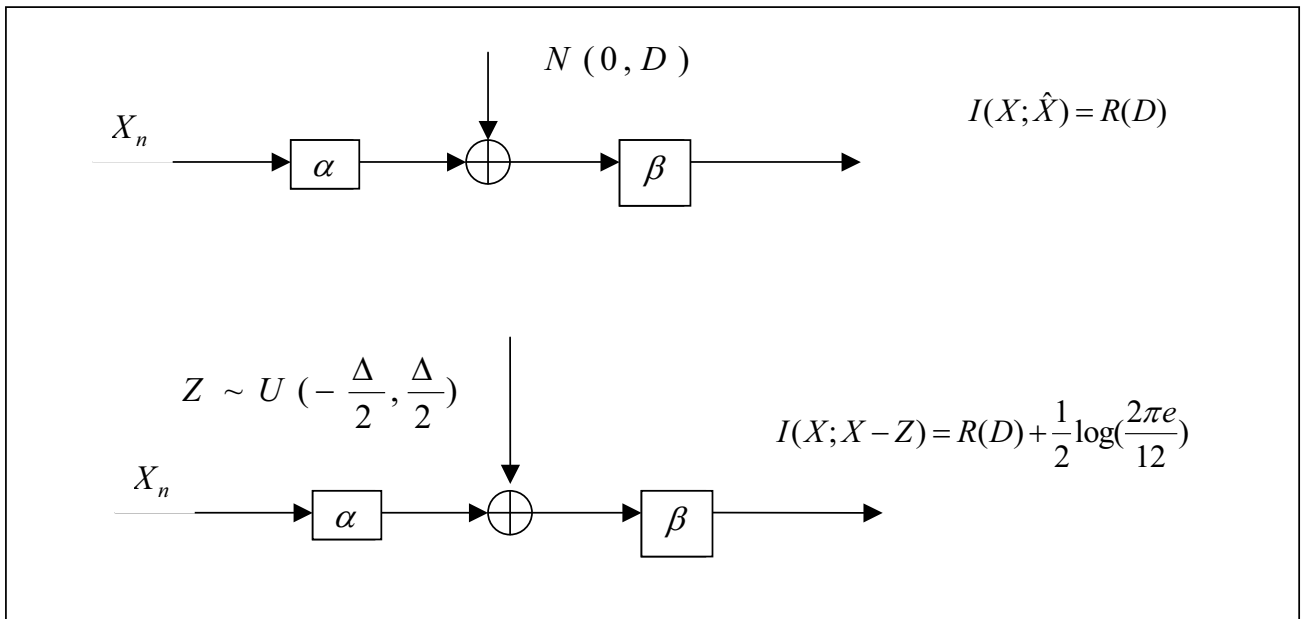
$$I(x; x - z) \cong h(x) - h(z) = h(x) - \log(\Delta) = h(x) - \frac{1}{2} \log\left(\frac{D}{12}\right) =$$

$$= R_{SLB}(D) + \frac{1}{2} \log\left(\frac{2\pi e}{12}\right) \approx R_{SLB}(D) + \frac{1}{4} \text{ bit}$$

היתירות ברזולוציה גבוהה קטנה מ-  $\frac{1}{4} \text{ bit}$ .

2.  $\text{redudancy} \triangleq H(Q|Z) - R(D) \cong \frac{1}{2} + \frac{1}{2} \log\left(\frac{2\pi e}{12}\right)$ . בכל רזולוציה.

3. ערוץ מגשים ביחס ל MSE של מקור גאוס:



הערה- ניתן להחליף כל  $\frac{1}{12}$  ב  $G_k$  ולהגיע לתוצאות של סריג.

## קידוד תהליכים גאוסיים

אנטרופיה – וקטור גאوسي  $X \sim N(0, R_x)$  אליפסואיד שמרכזו בראשית הצירים.

$$2^{h(x)} \sim \text{"נפח המקור"}$$

נבצע טרנספורמציה למקרה הלבן:

$$T^t T = R_x^{-1}$$

$$\Rightarrow \det(T^2) = \det(R_x^{-1}) = [\det(R_x)]^{-1}$$

$$X_w = Tx = R_x^{-\frac{1}{2}} x$$

$$\text{Cov}(X_w) = I_k = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \dots & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\Rightarrow h(X_w) = kh(X_{w_i}) = k \frac{1}{2} \log(2\pi e)$$

$$\Rightarrow \text{volume}(\circ) = (2\pi e)^{\frac{k}{2}}$$

$$\Rightarrow \text{vol}(\circ) = \text{Vol}(\text{ellipse}) \times \det(T) = \text{Vol}(\text{ellipse}) \times \det(R_x)^{-\frac{1}{2}}$$

$$\Rightarrow 2^{h(X_w)} = 2^{h(X) \times \sqrt{\frac{1}{\det(R_x)}}}$$

$$\Rightarrow h(X) = h(X_w) + \frac{1}{2} \log(\det(R_x)) = \frac{k}{2} \log(2\pi e \det(R_x)^{\frac{1}{k}}) =$$

$$= \sum \frac{1}{2} \log(2\pi e \lambda_i), \det(R_x) = \prod \lambda_i, \lambda_i \triangleq \text{eigen-values}$$

$$R[k] = E(X_n X_{n+k}) \text{ תהליך גאوسي סטציונארי}$$

$$\lim_{k \rightarrow \infty} \frac{1}{k} h(X^k) = \lim_{k \rightarrow \infty} \frac{1}{k} \sum_{i=1}^k \log(2\pi e \lambda_i^{(k)})$$

Toplitz Limit disturbance – ע"ע שואפים לצפיפות  $S_x(f)$  על ציר התדר  $f \in (-\frac{\pi}{2}, \frac{\pi}{2})$



$$\begin{aligned}
\bar{h}(X) &= \text{entropy - rate} = \lim_{k \rightarrow \infty} \frac{1}{k} h(x_1, \dots, x_k) = \\
&= \lim_{k \rightarrow \infty} \frac{1}{k} \sum_{i=1}^k \log(2\pi e \lambda_i^{(k)}) = \int_{-\frac{1}{2}}^{\frac{1}{2}} \log(2\pi e S_x(f)) df \leq \\
&\leq_{\text{Jensen}} \frac{1}{2} \log(2\pi e \sigma_x^2)
\end{aligned}$$

$$\left[ \sigma_x^2 = \frac{1}{k} \sum_{i=1}^k \lambda_i^{(k)} \rightarrow_{k \rightarrow \infty} \int_{-\frac{1}{2}}^{\frac{1}{2}} S_x(f) df \quad \text{משום ש:} \right]$$

פונקציית קצב-עיוות:

עבור וקטור גאوسي -  $X \sim N(0, R_x)$

$$R_x = P \begin{pmatrix} \lambda_1 & & 0 \\ & \dots & \\ 0 & & \lambda_k \end{pmatrix} P^t, PP^t = P^t P = I_k$$

נגדיר  $\tilde{x} = P^t x$

$$\begin{aligned} \text{cov}(\tilde{x}) &= E(P^t x (P^t x)^t) = E(P^t x x^t P) = \\ &= P^t R_x P = P^t (P \Lambda P^t) P = \Lambda \end{aligned}$$

← רכיבי  $\tilde{x}$  ח"ק ← בת"ס (גאוסי).

מה קורה לעיוות הריבועי כתוצאה מהטרנספורמציה? לא משתנה!

[ האנרגיה נשמרת בהתמרה אורתוגונלית – Parsaval's Theorem ]

$R_x(D)$  - פונקציית קצב עיוות תחת אילוף של עיוות D ממוצע לרכיביה

$$\begin{aligned} R_x(D) &= R_{\tilde{x}}(D) = \min_{\{D_1, \dots, D_k: \frac{1}{k} \sum D_k \leq D\}} \frac{1}{k} \sum_{i=1}^k R_{\tilde{x}_i}(D_i) = \\ &= \min_{\{D_1, \dots, D_k: \frac{1}{k} \sum D_k \leq D\}} \frac{1}{k} \sum_{i=1}^k \frac{1}{2} \log\left(\frac{\lambda_i}{D_i}\right) \Big|_{D_i \leq \lambda_i} \end{aligned}$$

פתרון לבעיית אופטימיזציה זו, מושג על ידי כלל מזיגת מים. נמלא מים בכמות KD מתחת לספקטרום

הע"ע של  $R_x$ .

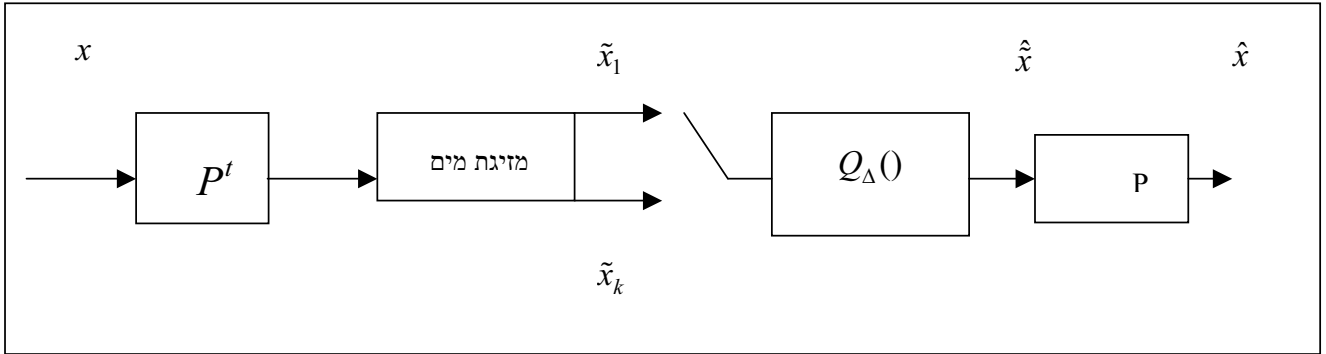
הערה - קל לראות שתחת ההצבה  $\forall i. D_i = D$  מתקבל:

$$R_x(D) \geq \frac{1}{k} \sum_{i=1}^k \frac{1}{2} \log\left(\frac{\lambda_i}{D}\right) = R_{SLB}(D)$$

ושוויון אדם  $D \leq \min_i \lambda_i$

השלכה למערכות קידוד:

א. מקור וקטורי – קידוד התמרה ( Transform Coding )



נשווה עם קוונטיזציה סקלארית :

$$rate \sim \frac{1}{2} \log\left(\frac{\sigma_x^2}{D}\right) = \frac{1}{2} \log\left(\frac{\frac{1}{k} \sum_{i=1}^k \lambda_i}{D}\right)$$

הרווח לעומת קוונטיזציה סקלארית:

$$"memory - gain" \geq \log\left(\frac{1}{k} \sum \lambda_i\right) - \frac{1}{k} \sum \log(\lambda_i)$$

$$R(D) = R_{SLB} \text{ אם } \dots$$

ב. תהליך סטציונרי – קידוד פסי-תדר ( sub-band coding )

המטרה – חלוקה שווה של העיוות בין התדרים.

D – העיוות "כמות המים".

$$B_i(\theta) = \text{inband} = \{f : S(f) > \theta\}$$

$$B_o(\theta) = \text{outband} = \{f : S(f) > \theta\}$$

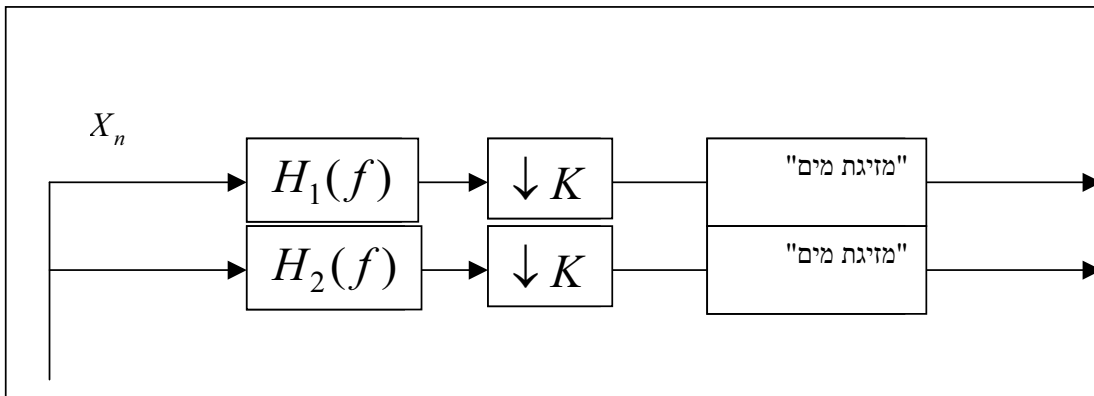
$$D = \theta B_i(\theta) + \int_{B_o(\theta)} S(f) df$$

זהו פתרון אידיאלי לפי שאנון ( תחת ההנחה שהקשר בין הקצב לעיוות הוא  $\frac{1}{2} \log \frac{S(f)}{D(f)}$  )

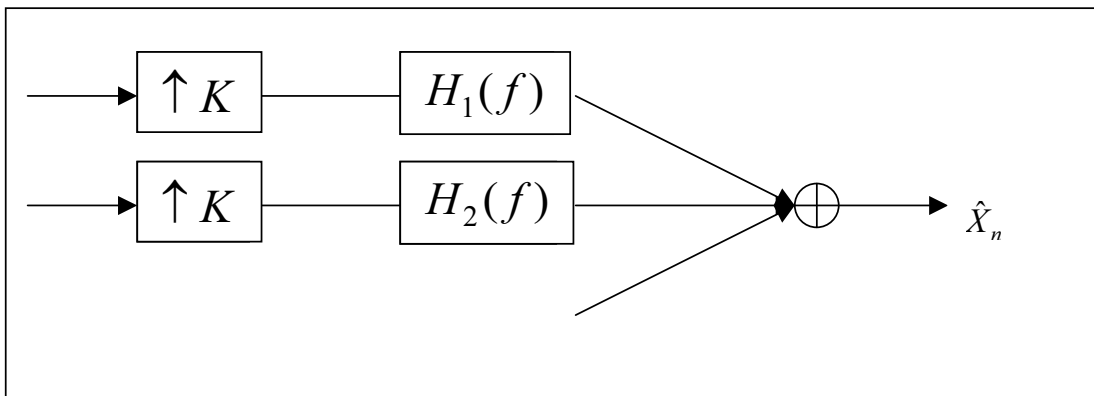
אבל בסכמה ממשית זה רק מייצג בקירוב

$$R(D) = \int_{-0.5}^{0.5} \frac{1}{2} \log \frac{S(f)}{D(f)} df = \int_{f \in B_i} \frac{1}{2} \log \frac{S(f)}{\theta} df$$

:Sub-band Coding



חלק מהביטים הם out-band ולא עוברים כלל, מה שכן עובר ממשיך לשלב הבא:



אידאלית היינו מעוניינים במסננים חדים בעלי תגובת תדר מלבנית אבל זה בלתי אפשרי ( במערכת סיבתית ). לכן, באופן מעשי, נשתמש במסנני QMF המבטיחים perfect reconstruction.

$$S(f_i) = \sigma^2; i = 1, \dots, k$$

$$D_i = \begin{cases} \theta, inband \\ \sigma^2, outband \end{cases}$$

רצינו להשיג:

התמרה מלכסנת ← חוסר קורלציה ← ( עבור גאוסי ) אי תלות ← קידוד נפרד של הרכיבים ← אופטימלי! אך שימוש ב QMF לא משיג זאת.

הערות:

1. "הקצאת ביטים" במובן של  $\sigma_i^2$  ( תחום גרעיני ) ו  $D_i$  ( מספר רמות קוונטיזציה ).
2. לרווח גרעיני ( space filling gain ) ← קוונטיזציה וקטורית לכל פס תדר.
3. ניתן להכליל למדד עיוות משוקלל בתדר:

$$D = \int w^2(f) S_{error}(f) df; error = \hat{x}_n - x_n$$

$$D_{weighted} = \sum_{i=1}^k w_i^2 E(\tilde{x}_i - \hat{\tilde{x}}_i)^2 = \sum_{i=1}^k E(w_i \tilde{x}_i - w_i \hat{\tilde{x}}_i)^2 =$$

$$= D_{MSE} w.r.t - \tilde{x} = W\tilde{x}$$

$\tilde{x}$  - במרחב האורתוגונלי.

4. עבור וקטורים ( transform coding ) מקובל להתמיר עם DCT – פתרון רובאסטי להתמרה מלכסנת של וקטורים שנגזרו ממקור מרקובי סטציונרי.

## References:

ECDQ –

J. Ziv, "On Universal Quantization" IEEE Transactions on Information Theory,  
Theory,  
pp. 344-347, May 1985

R.Zamir and M.Feder, "On Universal Quantization by Randomized  
Uniform/Lattice Quantizers", IEEE Transactions on Information Theory,  
pp. 428-436, March 1992

R(D) and water pouring –

Cover-Thomas book (1991)

for transform coding, sub-band coding –

Gibson-Berger book (1998)

# Compression under Uncertainty Conditions

## I Lecture Outline

There are several types of uncertainty. Corresponding to them, there are several information-theoretic formulations:

- Required quality of reconstruction  $\Rightarrow$ 
  - Successive Refinement (SR) (also Progressive Transmission)
- Channel capacity or noise level  $\Rightarrow$ 
  - Unequal Error Protection (SR+BCC)
  - Graceful Degradation (Analog vs. Digital Transmission)
- Network characteristics - Connectivity, Delays, Traffic loads  $\Rightarrow$ 
  - Multiple Descriptions
  - Rateless codes (not discussed)
- Correlative signal known only to the decoder (sensor array transmission)  $\Rightarrow$ 
  - Hybrid Analog+Digital Transmission Systems
  - Source coding with side information at the receiver (Wyner-Ziv coding)
  - Distributed source coding (lossy Slepian-Wolf, not discussed)
- Distortion measure (not discussed)  $\Rightarrow$ 
  - Robust codebooks
  - Side-information dependent distortion measures

## II Uncertainty: Required Quality

Different users may require different levels of reconstruction quality, and may want to improve quality on demand. For instance, transmitting a large compressed image via the internet may take some time (depending on the size of the image and the speed of connection), so in order for the user not to get bored the image can be progressively encoded, so that a coarse reconstruction of the image may be quickly available for display, and as further encoded data is received, a more refined image will be displayed.

### II.1 Successive Refinement (SR)

The Setting:

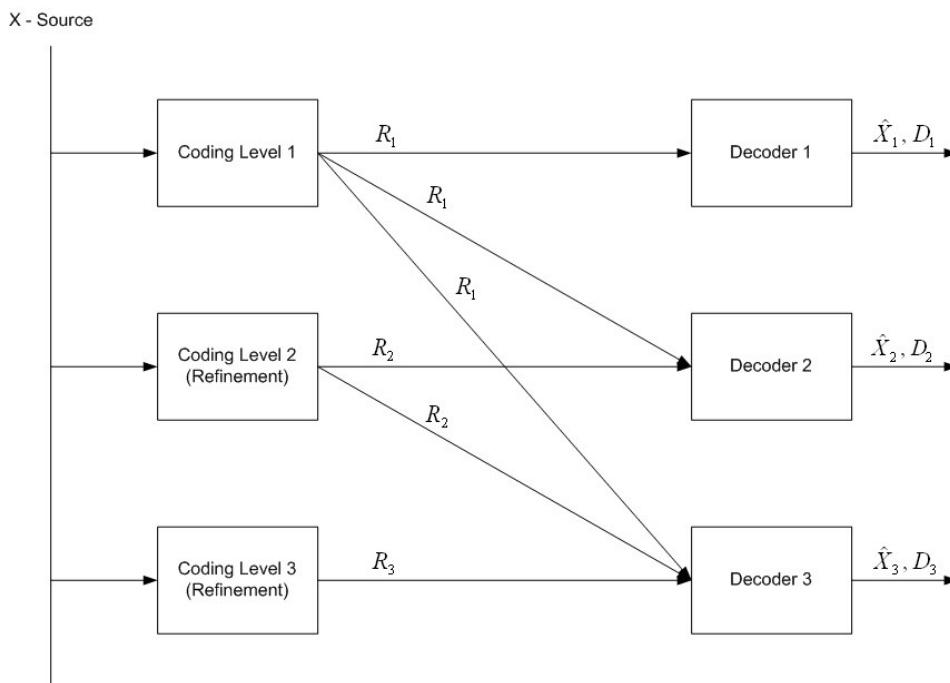


Figure 1: Successive Refinement System

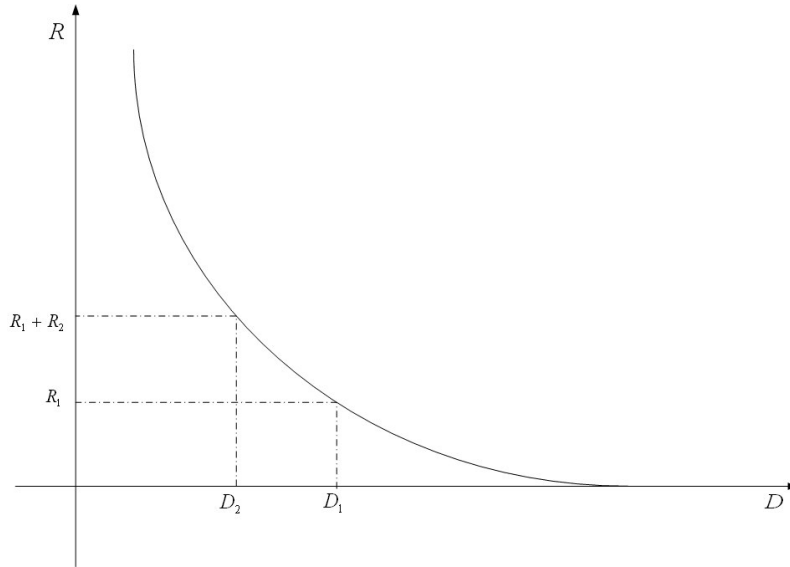
If all of the decoders work with the same distortion measure, then

$$D_1 \geq D_2 \geq D_3$$

Consider the case of two decoding steps. What are the rate pairs  $(R_1, R_2)$  allowing a desired distortion pair  $(D_1, D_2)$  ? The best we can hope for is given by the rate-distortion function:

$$\begin{aligned} R_1 &\geq R(D_1) \\ R_1 + R_2 &\geq R(D_2) \end{aligned} \tag{1}$$





and say that a source is *successively refinable* if the rates in (1) are attainable for every  $D_2 \leq D_1$ . In [1] it was proved that a necessary and sufficient condition for a source to be successively refinable is the existence of a conditional distribution  $p(\hat{x}_1, \hat{x}_2 | x)$  so that

$$\begin{aligned} Ed(X, \hat{X}_1) \leq D_1, \quad I(X; \hat{X}_1) &= R(D_1) \\ Ed(X, \hat{X}_2) \leq D_2, \quad I(X; \hat{X}_1, \hat{X}_2) &= R(D_2) \end{aligned}$$

and

$$p(\hat{x}_1, \hat{x}_2 | x) = p(\hat{x}_2 | x)p(\hat{x}_1 | \hat{x}_2) \tag{2}$$

where equation (2) is equivalent to saying that  $X, \hat{X}_1, \hat{X}_2$  can be written as a Markov chain  $X \rightarrow \hat{X}_2 \rightarrow \hat{X}_1$ , where  $\hat{X}_1$  is the coarser reconstruction of  $X$  and  $\hat{X}_2$  is the finer one. Notice that this also implies  $I(X; \hat{X}_1, \hat{X}_2) = I(X; \hat{X}_2)$ .

A practical system in the spirit of successive refinement is a multistage quantization system:

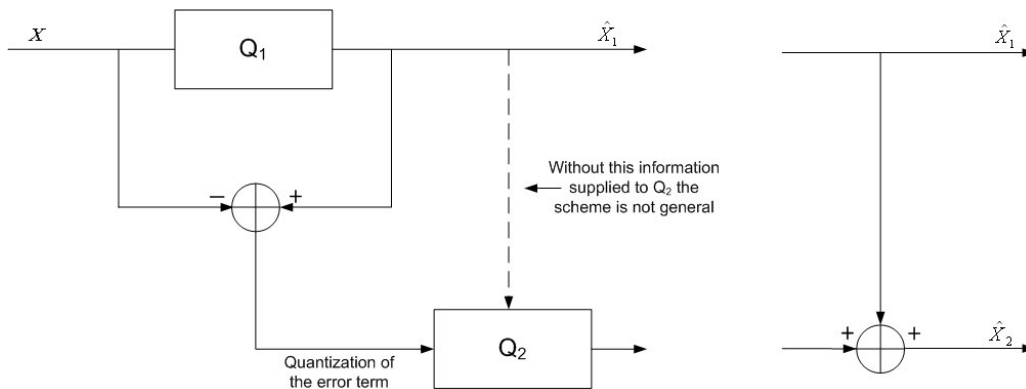


Figure 2: Multistage Quantization System

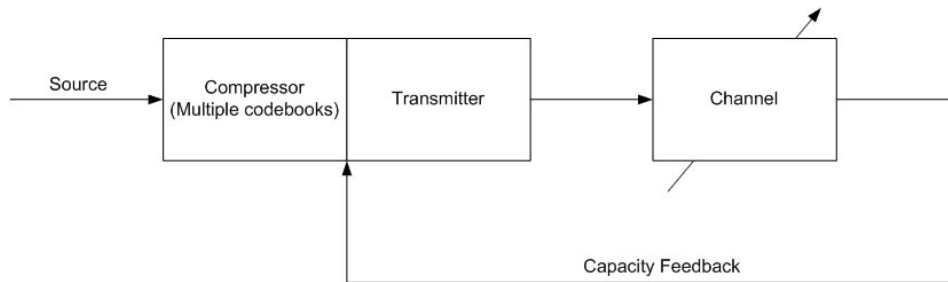
### III Uncertainty: Channel Capacity / Noise Level

Communicating a source over a channel whose capacity is unknown in advance may turn disastrous whenever the instantaneous capacity is too low, and possibly no decent reconstruction of the source can be guaranteed. In this section we describe two different approaches to this problem: One employs a source code and a channel code that enable some level of reconstruction dependent on the realized channel, and the other tries to guarantee that the distortion degradation when the capacity is somewhat lower than expected will not be severe.

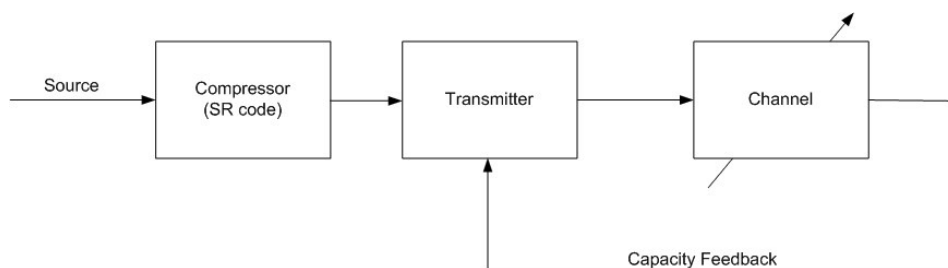
#### III.1 Unequal error protection

Consider several transmission scenarios:

1. Channel capacity is known to the transmitter and compressor  $\Rightarrow$  The compressor can match the codebook to the channel.



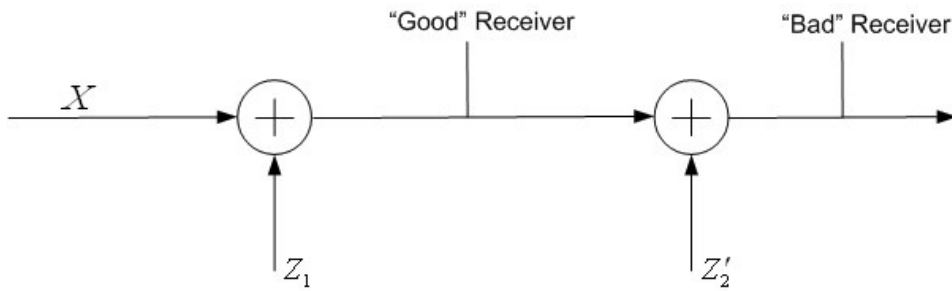
2. Channel capacity is known to the transmitter but not to the compressor (e.g., compression is performed on a remote computer)  $\Rightarrow$  The compressor can use a SR source code, the transmitter decides how many levels of refinement to use for transmission. Another practical (but less optimal) alternative is *transcoding*, where the compressor uses a single fixed codebook (not SR type) and the transmitter performs "recompression" according to the available capacity.



3. Channel capacity unknown to the transmitter and compressor  $\Rightarrow$  Separation principle (see remark at the end of the section): Concatenate a SR source code with a broadcast type

channel code. Every refinement level is suited to a virtual user that "sees" a channel with capacity that can accommodate the required rate. Effectively, the decoder uses the maximal number of refinement levels the instantaneous capacity allows, with high level refinements (dedicated to better receivers) acting as noise and reducing capacity. This is referred to as *unequal error protection*, since the higher refinement levels are less "protected" from noise in the channel, and will only be decoded by the better receivers.

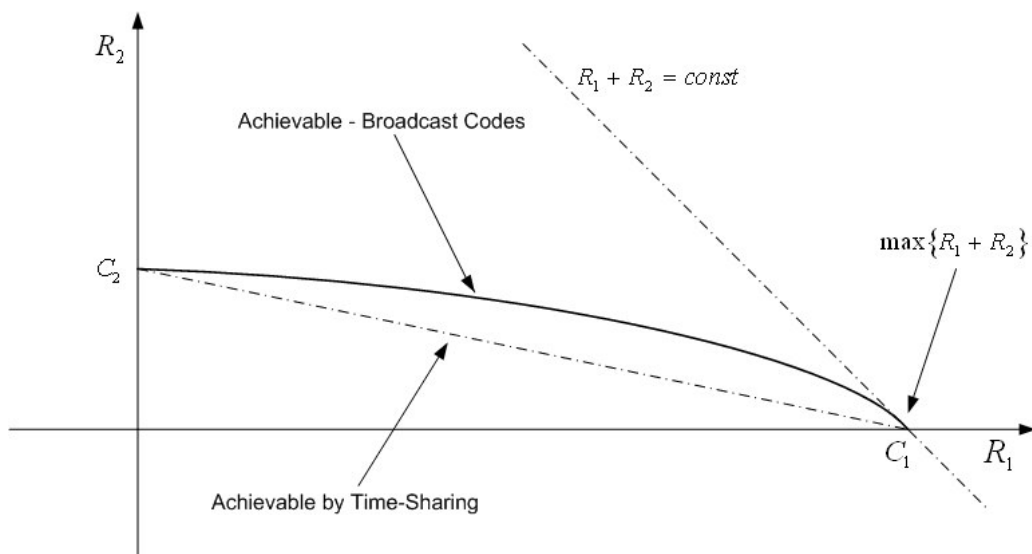
In a Gaussian channel where it is known that  $SNR \geq SNR_{min}$ , the transmitter can use a broadcast code suited for a degraded Gaussian broadcast channel. For instance, with two levels of refinement, the corresponding degraded broadcast channel for which the channel code is designed is



where

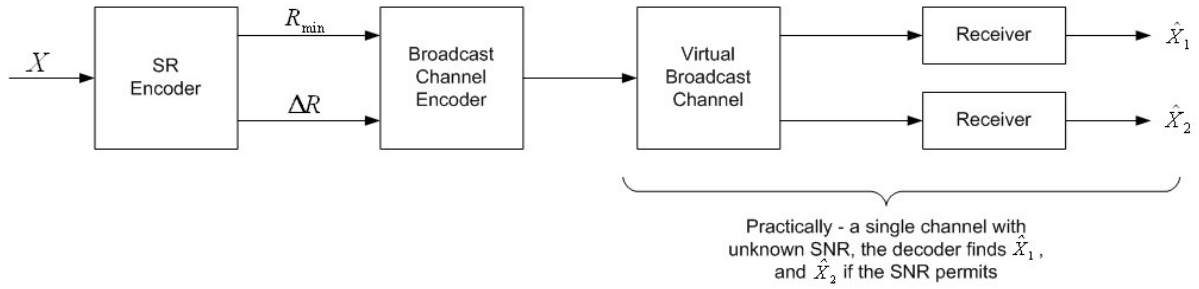
$$Z_1 \sim \mathcal{N}(0, \sigma_1^2), Z'_2 \sim \mathcal{N}(0, \sigma_2^2 - \sigma_1^2)$$

The capacity region for this channel is of the form



Notice there is an inherent trade-off between the rates. The higher the rate we want to dedicate to the bad receiver, the lower the rate we can provide to the good receiver.

concatenating the SR source code with the broadcast channel code we get



with an achievable distortion region of the form

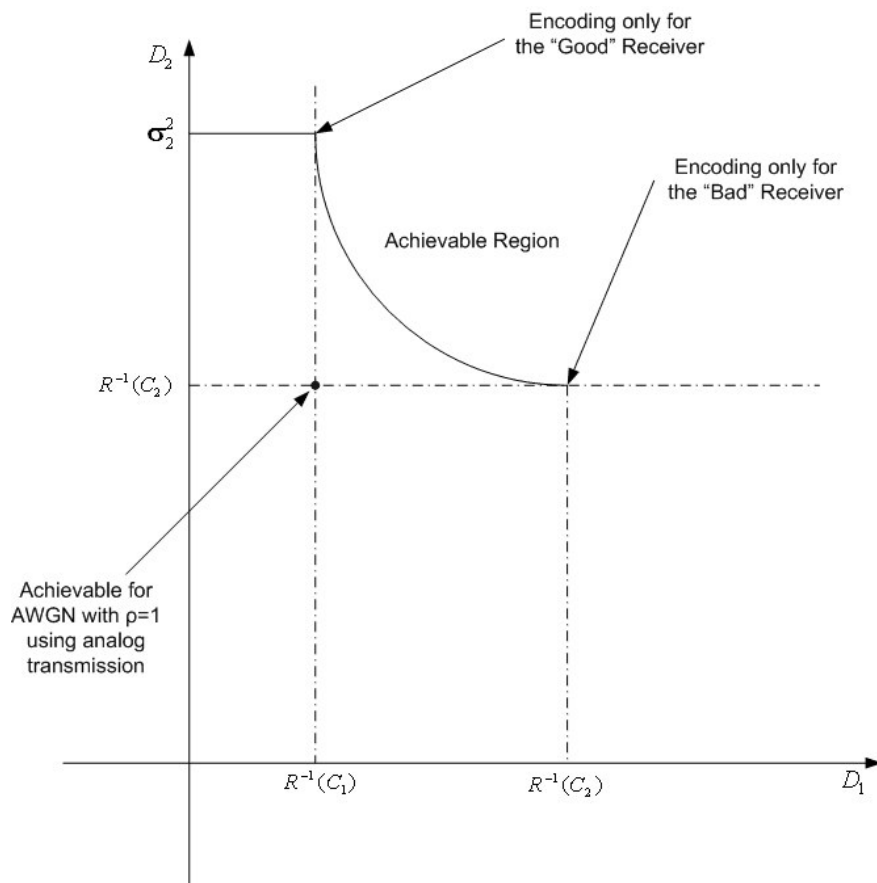


Figure 3: Achievable distortion region for SR+BCC

Notice that the separation principle does not apply here since this is not a point-to-point setting, and thus our suggested solution is not optimal. Indeed, as we shall see in the next subsection, in certain situations a better distortion region  $(D_1, D_2)$  can be attained using methods that are not purely digital.

### III.2 Graceful Degradation

Transmitting over a channel with unknown SNR (or to several users with different SNR's):

1. Limitations of pure digital transmission: The "threshold phenomena" - When transmission rate exceeds the instantaneous channel capacity, The block error probability tends to one.
2. Limitations of analog transmission (FM,SSB,AM):

Assume a flat Gaussian source with a quadratic distortion measure over a flat AWGN channel and no bandwidth expansion  $\rho \triangleq \frac{B_c}{B_s} = 1$ .

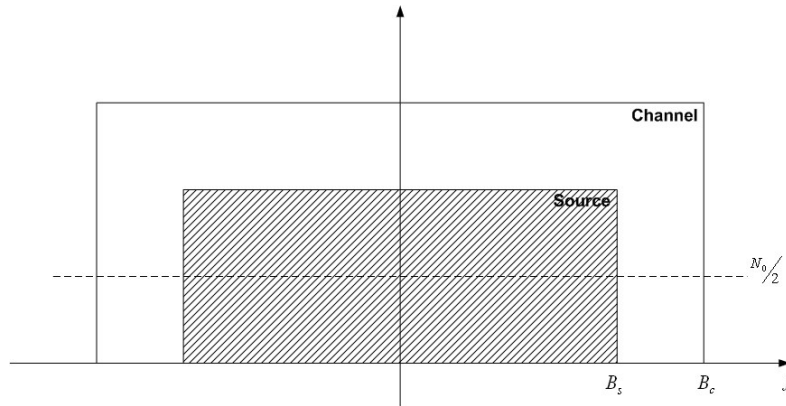
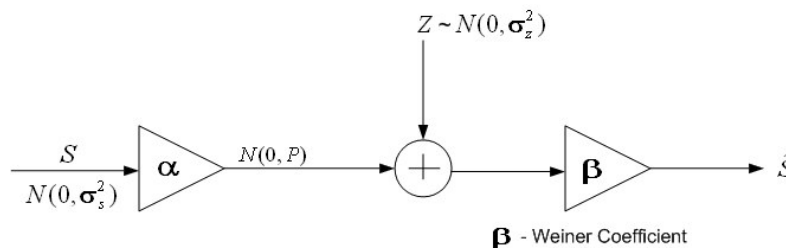


Figure 4: Flat Gaussian source over a flat AWGN channel

Fundamental limits:

$$R(D) = C \Rightarrow B_s \log \left( \underbrace{\frac{\sigma_s^2}{D}}_{SNR_{out}} \right) = B_c \log \left( 1 + \underbrace{\frac{P}{B_c N_0}}_{SNR_c} \right) \Rightarrow SNR_{out} = (1 + SNR_c)^\rho$$

Transmitting using SSB:



The input to output SNR relation is given by:

$$SNR_{out} = 1 + SNR_c$$

Since the channel is static, the Wiener factor  $\beta$  is assumed to be matched to the channel (i.e., the SNR is known to the receiver but not to the transmitter). We therefore see that analog transmission attains (for  $\rho = 1$ ) the ideal rate distortion point. Hence when the noise level increases, the distortion increases as well but in a graceful manner, and there is no threshold phenomena as in digital transmission (not true for FM).

For  $\rho > 1$  we can repeat the analog transmission in adjacent frequency bands and we get

$$SNR_{out} = 1 + \rho SNR_c$$

Which grows only linearly (rather than exponentially) with  $\rho$ , and if the constraint is on the total transmitted power, we have no gain from the bandwidth expansion at all. Hence we see that analog transmission does not take full advantage of bandwidth expansion, and similarly cannot approach capacity for a colored noise channel.

## IV Uncertainty: Network Characteristics

In many situations, the network through which the source is communicated may encounter unexpected traffic loads, delays, and packet losses. It is therefore essential that the user will be able to create a decent reconstruction of the source based on partial data only. Notice that a solution to this problem must be essentially different from progressive transmission (e.g., SR) as there is no guarantee to what part of the source description ends up at the user's terminal.

### IV.1 Multiple Descriptions (MD) (or Diversity Source Coding)

**Goal:** Describe the source using two (generally several) different "stand-alone" descriptions, so that possessing both provides a better reconstruction of the source. Useful for packet loss in internet communications.

The MD block diagram is depicted on the next page. Generally,

$$R_1 + R_2 \geq R(D_0)$$

We say that there is *no excess rate* when the above is satisfied with equality, meaning that the two descriptions "complement" each other so that there is no rate loss w.r.t. a single description attaining the same distortion  $D_0$ . In general, we cannot have both  $R_1 + R_2 = R(D_0)$  and  $R_i = R(D_i)$ ,  $i = 1, 2$ .

the MD problem is characterizing the achievable rate pairs  $(R_1, R_2)$  for a desired distortion

triplet  $(D_0, D_1, D_2)$ . It was shown [2] that any pair satisfying the conditions below is achievable:

$$\begin{aligned}
 R_1 &\leq I(X; U) \\
 R_2 &\leq I(X; V) \\
 R_1 + R_2 &\leq I(X; U, V) + \underbrace{I(U, V)}_{\text{Excess term between U and V}}
 \end{aligned}$$

for some pair of r.v.'s  $(U, V)$  (which represent the two descriptions) such that

$$\exists \hat{X}_1(U), \hat{X}_2(V), \hat{X}_0(U, V)$$

satisfying  $(D_0, D_1, D_2)$ . The full solution of this problem is however not yet available, except for the quadratic Gaussian case that was solved by [3].

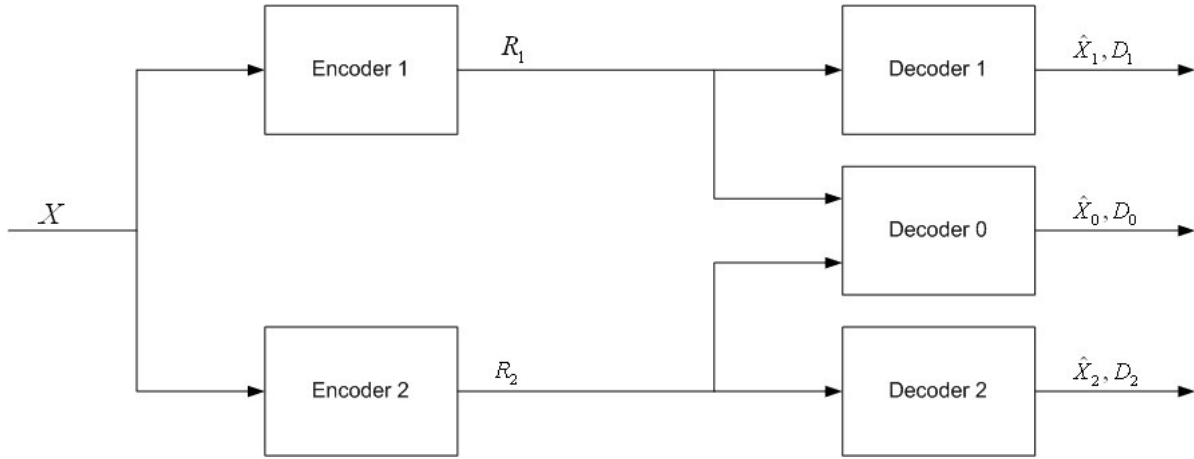


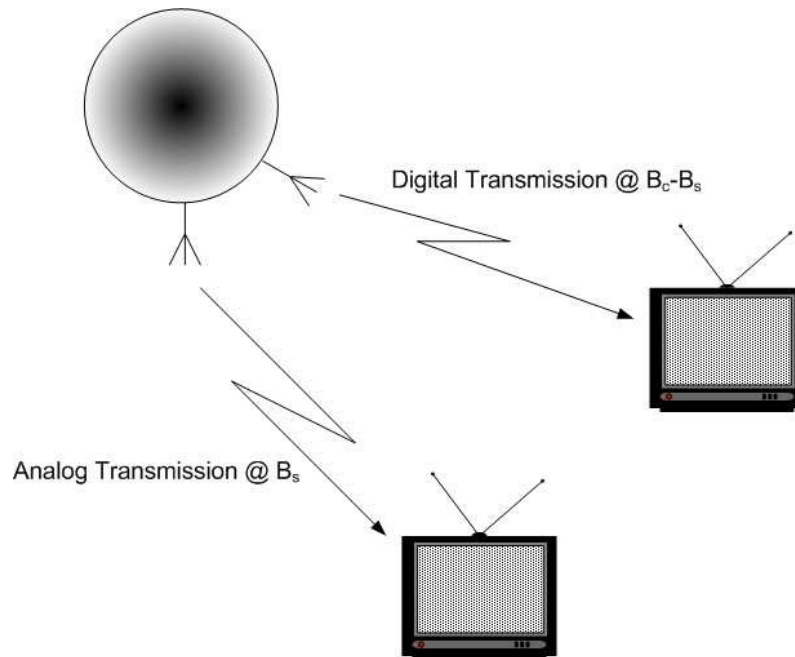
Figure 5: Multiple Descriptions System

## V Uncertainty: Correlative Signal Known Only to the Decoder

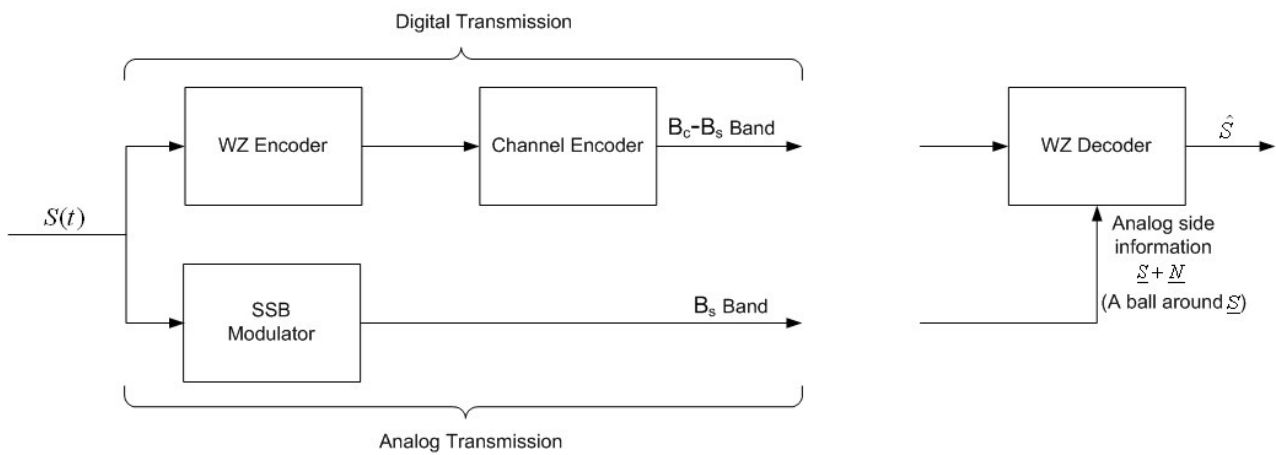
In this section we discuss the situation where the decoder has side information in the form of a signal correlative to the source, that is not available at the encoding terminal. We present settings where this may occur and means of utilizing this side information to attain a lower encoding rate.

### V.1 Hybrid Analog+Digital Systems

1. Systematic transmission (back-compatibility):



The decoder can use the analog data as side information. The encoder has no access to the received distorted analog signal, but nonetheless can encode the digital data to allow the decoder to use this side information. This refinement without a reference point is called *coding with side information at the receiver*, or *Wyner-Ziv coding*, and is discussed on the next subsection.



## 2. Analog Refinement:

Digital encoding is designed for the bad receiver, and analog refinement data is sent for the benefit of the better receivers, providing them with the graceful degradation property.



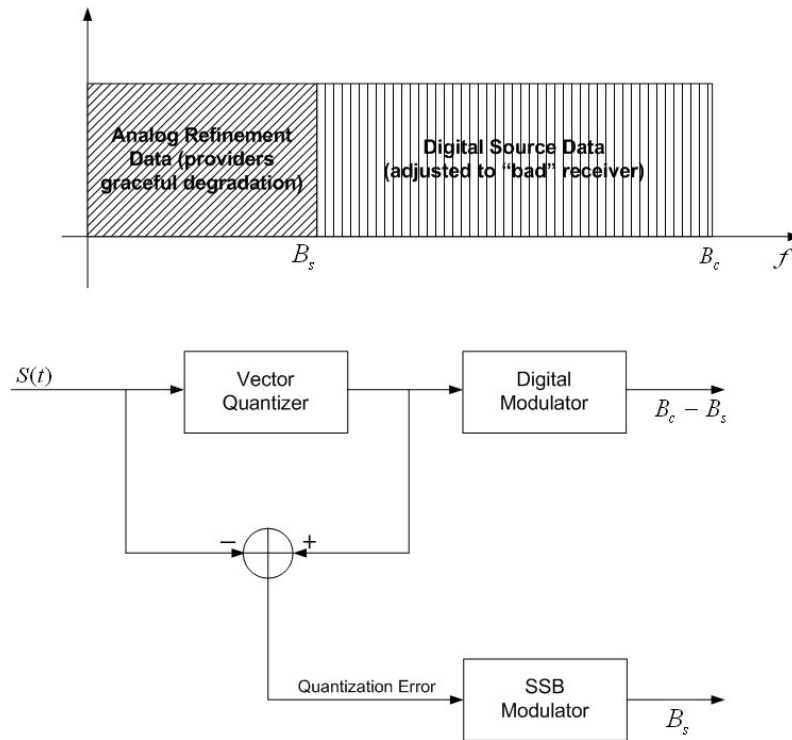
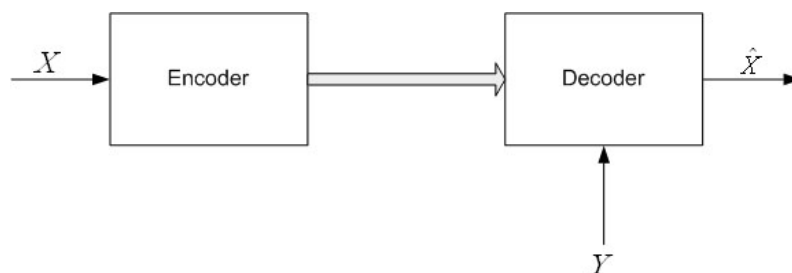


Figure 6: Hybrid System using Analog Refinement

## V.2 Wyner-Ziv (WZ) Coding

The Wyner-Ziv problem is one of encoding with side information available only to the decoder. It is desirable to utilize the existence of this side information to attain a lower encoding rate albeit the information is not available while encoding. The basic idea is demonstrated in the following riddle: Assume the ambient temperature is a nonnegative integer, that changes by exactly one degree (up or down) every day. Alice knows the temperature yesterday, and she wants to find out the today's temperature by receiving a single bit from her friend Bob, who knows it but already forgot yesterday's temperature. How is that possible? Solution: Bob transmits a zero if today's temperature divides by four, and one otherwise.

A Gaussian setting of the problem is depicted below:



$$X = Y + Z, \quad Z \perp Y, \quad X \sim \mathcal{N}(0, \sigma_x^2) \quad Z \sim \mathcal{N}(0, \sigma_z^2)$$

The achievable rate-distortion function  $R_{x|y}^{WZ}(D)$  satisfies

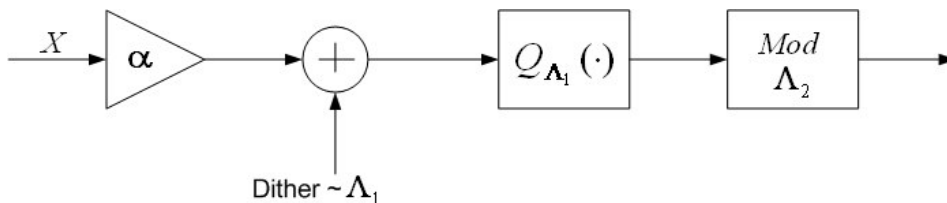
$$R_{x|y}(D) = \frac{1}{2} \log\left(\frac{\sigma_z^2}{D}\right) = R_{x|y}^{WZ}(D)$$

Generally however

$$R_{x|y}(D) \leq R_{x|y}^{WZ}(D)$$

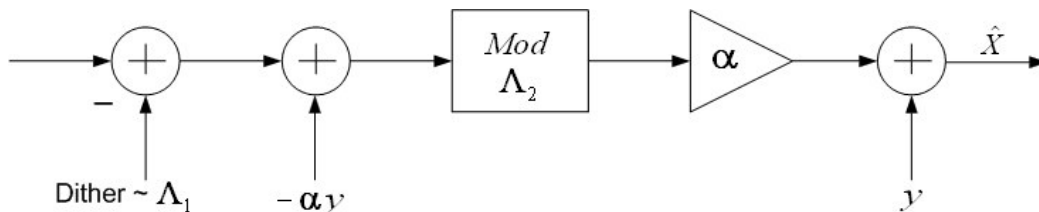
The proof [4] uses a *random binning* technique and is nonconstructive. We now described a constructive solution using *algebraic binning* [5].

Encoder's structure :



- $\Lambda_1$ : Fine lattice, volume determined according to the required distortion  $D$ .
- $\Lambda_2$ : Coarse lattice, volume determined according to the a-priori uncertainty at the decoder, i.e., proportional to  $\sigma_z^2$ .
- $\Lambda_2$  is nested in  $\Lambda_1$  ( $D < \sigma_z^2$ )
- $\alpha = \sqrt{1 - \frac{D}{\sigma_z^2}}$  is the Wiener coefficient.

Decoder's structure:



The rate attained by this scheme is

$$R = \frac{1}{K} \log\left(\frac{V_2}{V_1}\right) \approx \frac{1}{2} \log\left(\frac{\sigma_z^2}{D}\right) \quad (\text{for a large dimension } K)$$

Where  $V_i$  is the basic cell volume of  $\Lambda_i$ . Notice that for  $\alpha = 1$  we get  $\hat{x} = Q_{\Lambda_1}(x)$ .

**The basic idea:** Assume  $\alpha = 1$ . The encoder quantizes the source w.r.t.  $\Lambda_1$  (using dither to create a uniform distribution over the cell and satisfy the required distortion  $D$ ), performs a modulo- $\Lambda_2$  operation, and sends the result. The decoder now knows that the (quantized) source point is one of an (infinite-numbered) coset points of the received point. The decoder also knows  $Y$  and therefore knows that the correct point resides, with high probability, in a ball of radius  $\sigma_z^2$  around  $y$ , which corresponds to a cell of  $\Lambda_2$  centered at  $y$ . The decoder then places a lattice cell centered at  $y$ , and selects  $\hat{x}$  as the (single) coset point of the received point (w.r.t. the lattice  $\Lambda_2$ ), that intersects with that cell.

## References

- [1] W. H. R. Equitz and T. M. Cover, "Successive refinement of information," *IEEE Trans. Info. Theory*, vol. IT-37, pp. 269 - 275, March 1991.
- [2] A. A. El Gamal and T. M. Cover, "Achievable rates for multiple descriptions," *IEEE Trans. Info. Theory*, vol. IT-28, pp. 851 - 857, November 1982.
- [3] L. H. Ozarow, "On the source coding problem with two channels and three receivers," *Bell Labs Technical Journal*, vol. 59, pp. 1909 - 1922, July 1980.
- [4] A. D. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Info. Theory*, vol. IT-22, pp. 1 - 10, January 1976.
- [5] R. Zamir, S. Shamai, and U. Erez, "Nested linear/Lattice codes for structured multiterminal binning," *IEEE Trans. Info. Theory*, vol. IT-48, pp. 1250 - 1276, June 2002.