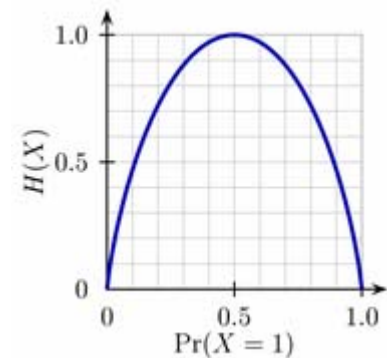


Information und Kommunikation

Hartmut Klauck
Universität Frankfurt
SS 07
14.5.



Feedback

- Wir betrachten nun gedächtnislose Kanäle mit *Feedback*
- Wir wollen zeigen, dass Feedback nicht zu größerer Kapazität führt
- Daher betrachten wir die stärkste Variante von Feedback: Die Ausgabe des Kanals Y_i wird an den Sender und den Empfänger geschickt

Feedback

- **Definition 9.1:**
 - Ein $(2^{Rn}, n)$ -Feedback-Code ist eine Folge von n Abbildungen $K(W, Y_1, \dots, Y_{i-1}) \rightarrow X$, und eine Dekodierungsabbildung D , wobei W aus $\{1, \dots, 2^{Rn}\}$ ist, und X das Eingabealphabet des Kanals, die Y_1, \dots, Y_{i-1} die Ausgaben des Kanals in Schritten $1, \dots, i-1$ sind
- **Definition 9.2:**
 - Die Rate eines Codes K , und Erreichbarkeit von Raten sind wie zuvor definiert
- **Definition 9.2**
 - Die Kapazität C_F eines Feedback-Kanals ist das Supremum aller erreichbaren Raten

Feedback

- **Theorem 9.3:**
 - Für alle gedächtnislosen Kanäle gilt:
 $C_F = C = \max_{p(x)} I(X:Y)$
- Offensichtlich gilt: $C_F \geq C$, da alle Raten $< C$ erreicht werden können, ohne Feedback zu benutzen
- Die andere Richtung bedeutet, zu zeigen, dass alle Raten $> C$ nicht erreichbar sind, selbst mit Feedback

Beweis

- Gegeben seien ein Kanal mit Matrix $p(y|x)$, einer Verteilung $p(x)$, für die die Kapazität C erreicht wird
- Gegeben sei auch ein Feedback-Code mit Rate R , d.h. ein $(2^{Rn}, n)$ -Feedback-Code
- Wir nehmen wieder an, dass Nachrichten uniform aus $\{1, \dots, 2^{Rn}\}$ gezogen werden
- Es gilt $H(W) = Rn$

Beweis

- Y_1, \dots, Y_n sei die Ausgabe des Kanals
- X_1, \dots, X_n die über den Kanal gesendeten Zeichen
- P_e die Fehlerwahrscheinlichkeit
- Es gilt:

$$\begin{aligned} nR &= H(W) = H(W|Y_1, \dots, Y_n) + I(W:Y_1, \dots, Y_n) \\ &\leq 1 + P_e nR + I(W:Y_1, \dots, Y_n) \text{ mit Fano} \end{aligned}$$

- Wir wollen nun $I(W:Y_1, \dots, Y_n)$ beschränken

Beweis

$$\begin{aligned} I(W:Y_1, \dots, Y_n) &= H(Y_1, \dots, Y_n) - H(Y_1, \dots, Y_n | W) \\ &= H(Y_1, \dots, Y_n) - \sum_{i=1}^n H(Y_i | W, Y_1, \dots, Y_{i-1}) \\ &= H(Y_1, \dots, Y_n) - \sum_{i=1}^n H(Y_i | W, Y_1, \dots, Y_{i-1}, X_i) \\ &\quad X_i \text{ ist eine Funktion von} \\ &\quad W, Y_1, \dots, Y_{i-1}, X_i \\ &= H(Y_1, \dots, Y_n) - \sum_{i=1}^n H(Y_i | X_i) \\ &\quad Y_i \text{ ergibt sich durch den} \\ &\quad \text{gedächtnislosen Kanal aus } X_i \end{aligned}$$

Beweis

Es gilt also

$$I(W:Y_1, \dots, Y_n)$$

$$= H(Y_1, \dots, Y_n) - \sum_{i=1}^n H(Y_i | X_i)$$

$$\leq \sum_{i=1}^n H(Y_i) - \sum_{i=1}^n H(Y_i | X_i)$$

$$= \sum_{i=1}^n I(Y_i: X_i)$$

$$\leq Cn$$

Beweis

- Insgesamt erhalten wir
- $R_n \leq 1 + P_e R_n + C_n$
- Oder $R \leq P_e R + C + 1/n$
- Wenn P_e gegen 0 geht, muss gelten, dass $R \leq C$
- D.h. $C_F = \sup R \leq C$

Bemerkung

- Es ist also nicht möglich, in Feedback Kanälen größere Raten zu erreichen, aber es kann einfacher sein, diese Raten zu erreichen
- Beispiel: Erasure Kanal, einfaches erneutes Senden erreicht die Kapazität des Kanals

Source/Channel Coding Theorem

- Wir haben bisher zwei Fragen unabhängig betrachtet:
 - Kompression von Daten
 - Senden der Daten über einen Kanal
- Wir erhielten folgende Bedingungen:
 - Kompression: $R > H$ (die Rate eines Codes muss größer als die Entropie der Quelle sein)
 - Kodierung: $R < C$ (Rate muss unterhalb der Kapazität liegen)
- Wenn wir beides miteinander kombinieren erhalten wir $H < C$, das heißt die Entropie der Quelle muss kleiner als die Kapazität sein

Source/Channel Coding Theorem

- Frage: ist die Bedingung $H < C$ hinreichend und notwendig, um Daten aus einer Quelle über einen Kanal zu senden?
- Frage: ist es optimal, Daten erst zu komprimieren (z.B. Huffman), und dann über einen Kanal zu senden (mit Kodierung der komprimierten Daten)
- Beispiel: Sprache
 - Quelle: Text
 - Kanal: Sprechen
 - Sprache scheint von sich aus „fehlertolerant“ zu sein
 - Ist es besser/gleich gut, die Sprache zu komprimieren, und dann fehlerkorrigierend zu kodieren?

Source/Channel Coding Theorem

- **Theorem 9.4**

- Seien V_1, \dots, V_n Zeichen, die unabhängig gemäß einer Verteilung q gezogen sind
- Gegeben sei weiterhin ein Kanal $p(y|x)$ mit Kapazität C
- Es gibt einen Code, der erlaubt V_1, \dots, V_n mit gegen 0 gehendem Fehler (mit Codeworten der Länge n) zu kommunizieren, wenn $H(V) < C$
- Wenn es einen Code (mit Codeworten der Länge n) mit gegen 0 gehendem Fehler gibt, gilt $H(V) \leq C$

Bemerkungen

- Dies kann (zum einen gewissen Grad) verallgemeinert werden in Bezug auf die Quelle (nicht nur unabhängige Wahl der Zeichen)
- „Stationäre ergodische Quellen“
- $H < C$ ausreichend folgt bereits aus unseren bisherigen Ergebnissen (6.9/7.3)

Beweis 9.4

- Gegeben seien n unabhängige Zeichen V_1, \dots, V_n (mit Alphabet $\{0, \dots, D-1\}$)
- Diese werden kodiert zu X_1, \dots, X_n , und über einen Kanal geschickt (Empfangen wird Y_1, \dots, Y_n)
- U_1, \dots, U_n sei die Ausgabe der Dekodierung
- Mit Fano gilt:
- $H(V_1, \dots, V_n | U_1, \dots, U_n) \leq 1 + P_e \log(D^n) = 1 + P_e n \log D$

Beweis 9.4

$$\begin{aligned} & H(V) \text{ (die Entropie der Quelle)} \\ & \leq H(V_1, \dots, V_n)/n \\ & = H(V_1, \dots, V_n | U_1, \dots, U_n)/n + I(V_1, \dots, V_n : U_1, \dots, U_n)/n \\ & \leq (1 + P_e n \log D)/n + I(X_1, \dots, X_n : Y_1, \dots, Y_n)/n \\ & \leq P_e \log D + 1/n + C \end{aligned}$$

Wenn P_e gegen 0 geht, muss $C \geq H$ gelten

Fehlerkorrigierende Codes

- Wir wollen nun praktikable Codes entwerfen
 - Gute Rate
 - Leicht/Schnell zu kodieren
 - Leicht/Schnell zu dekodieren
- Wir interessieren uns besonders für den symmetrischen Kanal (und später den Erasure Kanal)

Fehlerkorrigierende Codes

- Insbesondere wechseln wir etwas den Blickpunkt:
 - Wir konstruieren z.B. Codes der Länge n , welche d Fehler korrigieren können,
 - d.h. wenn das Codewort an höchstens d Stellen verfälscht ist, können wir es „wiederherstellen“
 - Der binäre symmetrische Kanal wird z. Beispiel erwartet pn von n Zeichen verfälschen, und mit hoher Wahrscheinlichkeit nicht mehr als $(p+\varepsilon)n$
 - Es reicht daher, $(p+\varepsilon)n$ Fehler korrigieren zu können, um mit hoher Ws. zu dekodieren

Fehlerkorrigierende Codes

- Wir betrachten zuerst den einfachsten fehlerkorrigierenden Code
- Angenommen wir wollen Bits senden
 - Wiederhole jedes Bit 3 mal
 - Dekodiere durch Mehrheitsentscheidung.
- Klar: Der Code hat Länge 3, und korrigiert einen Fehler
- Die Rate ist damit $1/3$
- Geht es besser?

Fehlererkennende Codes

- Eine Idee: Parity checks
 - Gegeben einen string x_1, \dots, x_n
 - Wir hängen ein Bit $\bigoplus_{i=1}^n x_i$ an
 - Wenn es nur einen Fehler gab, ist nachher das parity Bit mit Sicherheit falsch
- Frage: wie können wir den Fehler *korrigieren*

Hamming Codes

- Der 3-Repetition Code kodiert Bits gegen 1 Fehler mit Rate 1/3
- Wir konstruieren einen Code wie folgt:
- Setze

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

- Für eine Sequenz von Bits betrachten wir jeweils Blöcke von 4 Bits
- 4 Bits $b = b_1, \dots, b_4$ werden durch bG kodiert (Matrix-Vektor-Multiplikation über Z_2 , b ist ein Zeilenvektor)

Hamming Codes

- Der sich ergebende Code bildet also jeweils 4 auf 7 Bits ab, d.h. die Rate ist $4/7$
- Wir haben jetzt zu zeigen, dass wir einen Fehler pro Block korrigieren können

Hamming Codes

- **Behauptung:**
 - Wenn $b \neq b'$, dann stimmen bG und $b'G$ in ≥ 3 Komponenten nicht überein
- Dies bedeutet, dass bG mit einem Fehler immer noch 2 „falsche“ Komponenten bzgl. $b'G$ hat, und daher das „nächstliegende“ bG die korrekte Dekodierung ist

Hamming Codes

- **Definition 9.5:** Die Hamming Distanz $d_H(x,y)$ zwischen strings x und y ist die Anzahl der Positionen, wo diese nicht übereinstimmen
- **Bemerkung:** Dies ist eine Metrik
 - $d_H(x,y)=0$ gdw. $x=y$
 - $d_H(x,y) \leq d_H(x,z)+d_H(z,y)$
 - $d_H(x,y)=d_H(y,x)$
- Wenn also $d_H(bG,b'G) \geq 3$, dann kann 1 Fehler korrigiert werden

Codes

- Wir definieren nun einige Begriffe
- **Definition 9.6**
 - Ein Fehlerkorrigierender Code über einem Alphabet A ist eine Menge von Codeworten $C \subseteq A^n$
 - Die Distanz $\Delta(C)$ eines Codes ist die minimale Hamming Distanz zwischen zwei Codeworten
 - Ein Code kann t Fehler entdecken, wenn es möglich ist, zu entscheiden ob es mindestens 1 Fehler gab, wenn es höchstens t Fehler gab
 - Ein Code kann t Fehler korrigieren, wenn zu jedem Codewort, das an $\leq t$ Positionen abgeändert wird, das originale Codewort wiederhergestellt werden kann

Codes

- **Lemma 9.7**

$\Delta(C) = 2t+1$ genau dann wenn C $2t$ Fehler erkennt und t Fehler korrigiert

Beweis 9.7

$\Delta(C)=2t+1$:

- Jedes Codewort, an dem $\leq 2t$ Positionen geändert werden, ergibt kein Codewort
 - Jedes Codewort x an dem $\leq t$ Positionen geändert werden, so dass sich y ergibt, erfüllt $d_H(x,y) < d_H(y,z)$ für alle Codeworte z
- Umgekehrt: später