Felrättande koder

Antag att en följd av nollor och ettor ska skickas genom en kanal:
0 1 1 1 0 1 0 0 0 1 0 1 1...
Om det finns en viss risk (sannolikhet) för fel kanske vi får ut:
0 1 1 0 0 1 0 1 0 1 0 1 1...
Hur kan man rätta till felen med så lite jobb som möjligt? Detta är ett problem som jämt måste hanteras i all modern kommunikation.

Idé 1: Skicka alla siffror dubbelt:
00 11 11 11 00 11 00 00 00 11 00 11 11...
Vi kanske får ut
00 11 11 10 00 11 00 01 00 11 00 01 11...
Här indikerar 01 eller 10 att något blivit fel i överföringen men vi vet inte om ursprungliga siffran borde vara 0 eller 1. Vår tolkning blir
0 1 1 ? 0 1 0 ? 0 1 0 ? 1..., inte helt lyckat.

Idé 2: Skicka alla siffror trippelt:
000 111 111 111 000 111 000 000 000 111 000 111 111...
Vi kanske får ut
000 111 111 110 000 111 000 001 000 111 100 101 111...
som vi genom majoritetsbeslut tolkar som
0 1 1 1 0 1 0 0 0 1 0 1 1...
vilket är rätt följd!

En siffra 1 som skickas trippelt som 111 kan om det uppstår två fel, t.ex. om vi får ut 100 tolkas som 0. Hur stor är risken för detta? Om sannoliketen för att en bit eller siffra överförs korrekt är p (och vi antar att fel uppstår oberoende - kanske inte alltid sant) blir sannoliketen för rätt slutresultat

$$P_r = P(\text{rätt överföring}) = P(3 \text{ rätt}) + P(2 \text{ rätt}) = p^3 + 3p^2(1-p) = 3p^2 - 2p^3$$

P står för sannolikhet (probability)
Om t.ex. $p = 0.9$, d.v.s. 10% felrisk per bit, blir

$$P_r = 3 \cdot 0.9^2 - 2 \cdot 0.9^3 = 0.972$$

Risken för fel i överföring per siffra har minskat från 10% till 2.8% genom att skicka den trippelt. Priset är att vi måste skicka tre gånger så många bitar som i originalföljden.

Att överföra 1 som 111 med sannolikhet $p = 0.9$ att varje bit blir korrekt ger alltså sannolikheten $P_r = 0.972$ att 1 överförs korrekt.

Om vi vill överföra 4 siffror utan kontrollbitar blir sannolikheten att alla 4 överförs korrekt $p^4 = 0.9^4 \approx 0.66$. Med systemet att skicka varje trippelt blir sannolikheten att alla 4 överförs korrekt $P_r^4 = 0.972^4 \approx 0.89$.

För 7 siffror fås $p^7 = 0.9^7 \approx 0.48$ och $P_r^7 = 0.972^7 \approx 0.82$.

Finns bättre system?
Samma/mindre felrisk utan att behöva överföra allt trippelt?

Gruppera siffrorna som ska överföras i block med 4 siffror i varje, t.ex.
$0\ 1\ 1\ 1\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 1\ 1\ldots$
blockindelas som
$0\,1\,1\,1 \quad 0\,1\,0\,0 \quad 0\,1\,0\,1 \quad 1\ldots$

Till varje block läggs 3 kontrollsiffror så totalt 7 bitar överförs. Det sker genom att $1\,0\,0\,0$ görs till $1\,0\,0\,0\,1\,0\,1$ (sista $1\,0\,1$ är kontrollsiffrorna),

$0\,1\,0\,0 \longmapsto 0\,1\,0\,0\,1\,1\,0$,
$0\,0\,1\,0 \longmapsto 0\,0\,1\,0\,1\,1\,1$ och
$0\,0\,0\,1 \longmapsto 0\,0\,0\,1\,0\,1\,1$.

Dessa fyra 4-block bildar bas och andra block är linjärkombinationer. Kodning sker nu genom att vi säger att det ska vara en linjär avbildning. Om vi sätter blocken i kolonner så hamnar basblockens bilder som kolonner med längd 7 i en 7×4 - matris som beskriver kodningen, d.v.s. hur alla 4-block görs om till 7 bitar som ska överföras.

Avbildningsmatrisen blir $A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$ och t.ex. blocket $0\ 1\ 1\ 1$ ger

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

Alltså $0\ 1\ 1\ 1 \longmapsto 0\ 1\ 1\ 1\ 0\ 1\ 0$

Vi har räknat med reglerna (”modulo 2 ”):
$0+0=0,\ 0+1=1,\ 1+1=0,\ 0\cdot 0=0,\ 0\cdot 1=0,\ 1\cdot 1=1$

Koden ovan kallas Hamming(7,4)-koden.

Om man beräknar vad alla de 16 möjliga blocken avbildas på får man

0 0 0 0 0 0 0 , 1 0 0 0 1 0 1 , 0 1 0 0 1 1 0 , 0 0 1 0 1 1 1 , 0 0 0 1 0 1 1 , 1 1 0 0 0 1 1 ,
1 0 1 0 0 1 0 , 1 0 0 1 1 1 0 , 0 1 1 0 0 0 1 , 0 1 0 1 1 0 1 , 0 0 1 1 1 0 0 , 1 1 1 0 1 0 0 ,
1 1 0 1 0 0 0 , 1 0 1 1 0 0 1 , 0 1 1 1 0 1 0 , 1 1 1 1 1 1 1

Observation: alla skiljer sig på minst 3 platser. Om det vid överföring uppstår högst ett fel på de 7 bitarna kan vi återskapa originalblocket. Antag att $0\,1\,1\,1 \longmapsto 0\,1\,1\,1\,0\,1\,0$ överförs med ett fel till $0\,1\,1\,0\,0\,1\,0$. Den enda av de 16 uttrycken ovan som bara skiljer sig på högst en plats från detta är $0\,1\,1\,1\,0\,1\,0$ och vi drar den korrekta slutsatsen att det ursprungliga blocket var $0\,1\,1\,1$. Det spelar ingen roll om felet är i 4-blocket eller kontrollbitarna.

Sannolikheten att blocket överförs korrekt är (7 eller 6 bitar ska vara korrekta):
$P_B = P(7 \text{ rätt}) + P(6 \text{ rätt}) = p^7 + 7p^6(1-p) = 7p^6 - 6p^7 \approx 0.85$ om $p = 0.9$.

För $p = 0.9$ har vi alltså 85% chans att ett block överförs korrekt, och det kostar $7/4$ av vad det kostar att inte skicka kontrollbitar.

Utan kontrollbitar var sannolikheten att hela blocket överförs korrekt ca 66%.

Om varje siffra skickas trippelt var sannolikheten ca 89%, och det kostar 3 gånger så mycket.

Man kan designa blockkoder (Hammingkoder) med större block och förbättrad prestanda. Med 7 siffror i ett block och 8 kontrollbitar skiljer sig alla bilder på minst 5 platser och 2 fel kan rättas. Med $p = 0.9$ överförs ett 7-block korrekt med sannolikheten

$$P(\text{15 rätt}) + P(\text{14 r.}) + P(\text{13 r.}) = p^{15} + 15p^{14}(1-p) + 105p^{13}(1-p)^2 \approx 0.82$$

och det kostar $15/7$ av vad det kostar att inte skicka kontrollbitar.

Utan kontrollbitar är sannolikheten att hela blocket överförs korrekt $p^7 \approx 0.48$.

Om varje siffra skickas trippelt är sannolikheten $P_r^7 \approx 0.82$, och det kostar 3 gånger så mycket.

Avslutande kommentarer

Felrättande koder används vid all digital dataöverföring.

Många typer av koder finns, inte bara de linjära blockkoder vi beskrivit.

I tillämpningar är det ofta mycket stora datamängder som överförs och mycket liten andel fel tolereras.

I stället för vektorrum över $\mathbb{R}$ och vektorer i $\mathbb{R}^n$ har vi använt vektorrum över $\mathbb{Z}_2$, som består av bara 0 och 1 med räknereglerna ovan, och vektorer i $(\mathbb{Z}_2)^n$.

I stället för $A\bar{x} = \text{matris} \cdot \text{kolonn}$ används ofta inom kodning det ekvivalenta $(A\bar{x})^t = \bar{x}^t A^t = \text{rad} \cdot \text{matris}$.

Liknande matematik används i kryptering vid skickande av hemliga meddelanden. Man kan studera mer i olika ISY-kurser. Extra bra matematisk grund fås genom studier av (abstrakt) algebra, geometri, talteori, ...