Hoofdstuk 3

Toepassing: Codes

Als toepassing van vectorruimten over eindige lichamen kijken we naar foutenverbeterende codes. We benutten slechts elementaire kennis van vectorruimten, en van de volgende functie.

Definitie 3.1 Als R een commutatieve ring met 1 is en R^n de verzameling van rijtjes elementen van R ter lengte n (voor een $n \geq 1$), dan is de afbeelding $\langle\ ,\ \rangle : R^n \times R^n \to R$ gedefinieerd door aan twee elementen $v, w \in R^n$ het element $\langle v, w\rangle = \sum_{i=1}^{n} v_i w_i \in R$ toe te voegen

Opmerking 3.2 In het speciale geval dat R het lichaam $\mathbb{R}$ van de reële getallen is, is $\langle\ ,\ \rangle$ het *standaardinproduct*. Inproducten hebben allerhande belangrijke eigenschappen (op reële en complexe vectorruimten) die we in een volgend hoofdstuk in meer detail bekijken.

Het is voor enkele van de voorbeelden in dit hoofdstuk prettig deze algemenere definitie te hebben.

Definities 3.3 Een *lineaire (foutenverbeterende) code over* $\mathbb{F}_q$ is een lineaire deelruimte van $\mathbb{F}_q^n$, voor een eindig lichaam $\mathbb{F}_q$ en een $n \geq 1$. We zullen meestal simpelweg over een *code* $\mathcal{C}$ over $\mathbb{F}_q$ spreken. De *codewoorden* zijn de elementen van $\mathbb{F}_q^n$ die bevat zijn in $\mathcal{C}$. De *dimensie* $\dim \mathcal{C}$ van de code is de dimensie van de deelruimte. De *lengte* van de code is de dimensie n van de ruimte waaruit de vectoren afkomstig zijn, dus de lengte van de codewoorden.

Opmerkingen 3.4 Merk op dat het nulwoord $(0, 0, \ldots, 0) \in \mathbb{F}_q^n$ altijd in een code $\mathcal{C}$ zit. Het aantal elementen van de code $\mathcal{C}$ is gelijk aan q^k, als $k = \dim \mathcal{C}$ en q het aantal elementen van het lichaam. De verhouding tussen k en n wordt wel de 'rate' van de code genoemd: van de n coördinaten van elk codewoord zijn er k bij onafhankelijke informatiedragers, en de overige $n - k$ zijn 'opvulsel', redundantie. Het doel van *codetheorie* is om die $n - k$ extra coördinaten efficiënt te gebruiken om ervoor te zorgen dat de informatie uit het codewoord nog te achterhalen is wanneer een deel van het codewoord onderweg verminkt of verloren is geraakt, bijvoorbeeld door 'ruis' op de transmissielijn. Het uitgangspunt van het model dat ten grondslag ligt aan codetheorie is steeds dat de kans klein is dat informatie verminkt overkomt.

Voorbeelden 3.5 Het eenvoudigste voorbeeld van een foutenverbeterende code is de *repetitiecode* ter lengte n over $\mathbb{F}_2$: om een *bit* informatie (0 of 1) over te sturen wordt die bit n maal verzonden. De codewoorden zijn hier dus slechts de vectoren

$(1,1,\ldots,1)$ en $(0,0,\ldots 0)$ in $\mathbb{F}_2^n$, en de dimensie k is 1. Wanneer de ontvanger nu het woord $(1,1,0,1,1)$ ontvangt terwijl de binaire repetitiecode van lengte 5 gebruikt wordt, zal onder de aanname dat fouten vrij zelden op treden, de conclusie moeten luiden dat hoogstwaarschijnlijk het codewoord $(1,1,1,1,1)$ werd uitgezonden, maar dat op de derde positie een fout optrad. Ook wanneer er onderweg de uitzonderlijke pech optreedt dat twee bits worden omgezet kan de ontvanger hier nog tot de juiste conclusie komen met betrekking tot de bedoelde waarde van de bit: de ontvanger laat simpelweg de meerderheid van de ontvangen bits de doorslag geven. De repetitiecode ter lengte n is dus $(n-1)/2$ *foutenverbeterend* als n oneven is: we komen tot de juiste conclusie over het bit dat overgezonden moest worden mits de kans op i fouten maar (veel) kleiner is dan de kans op $n-i$ fouten.

Deze aanname, dat elk cijfer van de code met kleine waarschijnlijkheid onderweg wordt veranderd, en dat het uitgezonden codewoord de vector zal zijn geweest die in de code zit en op het kleinst mogelijke aantal plaatsen van de ontvangen vector afwijkt, leidt tot de methode om de informatie te achterhalen die *maximale waarschijnlijkheidsdecodering* wordt genoemd.

Definities 3.6 Het *gewicht* van een vector v uit $\mathbb{F}_q^n$ is het aantal posities i waarvoor de coördinaat $v_i \in \mathbb{F}_q$ niet nul is. De *Hammingafstand* $h(v,w)$ tussen twee vectoren v, w uit $\mathbb{F}_q^n$ is het aantal posities j waarop de coördinaten v en w verschillen: $h(v,w) = \#\{j : \ 1 \leq j \leq n \mid v_j \neq w_j\}$, dus $0 \leq h(v,w) \leq n$. De *minimumafstand* van een code $\mathcal{C}$ is het kleinste positieve getal d waarvoor codewoorden $v, w \in \mathcal{C}$ bestaan met $h(v,w) = d$.

Definitie 3.7 Een code $\mathcal{C}$ van lengte n over $\mathbb{F}_q$ heet *e-foutenverbeterend* als geldt dat voor elke vector $v \in \mathbb{F}_q^n$ er hoogstens één codewoord $c \in \mathcal{C}$ is met $h(c,v) \leq e$. Dat betekent dat uit een ontvangen vector v, die is ontstaan doordat in een codewoord op hoogstens e posities een verminking heeft plaatsgevonden, dit codewoord herleid kan worden. De code heet dan bovendien *$e+1$-foutendetecterend* als geldt dat elke op $e+1$ plaatsen verminkte vector afstand minstens $e+1$ tot alle codewoorden heeft. Het is mogelijk dat er dan meerdere codewoorden op afstand $e+1$ liggen, zodat unieke decodering onmogelijk is, maar er liggen in ieder geval geen codewoorden op afstand $\leq e$.

Voorbeeld 3.8 (parity check) Een simpel voorbeeld van een code die 1-foutdetecterend is, maar niet foutenverbeterend, is een code van lengte (zeg) 3 over $\mathbb{F}_2$ die bestaat uit alle woorden van even gewicht. Wanneer er dan een woord van oneven gewicht wordt ontvangen is het duidelijk dat er minstens 1 fout is opgetreden, maar wáár is niet altijd duidelijk.

Hieronder geven we nog drie voorbeelden van het gebruik van 'parity check's, in de zin dat er precies 1 controle-cijfer aan de informatie wordt toegevoegd die er voor zorgt dat er aan een enkele lineaire conditie is voldaan. Let wel op: in die voorbeelden gaat het niet meer over codes over $\mathbb{F}_2$, en in feite niet eens steeds over foutenverbeterende codes volgens onze definitie, want de onderliggende ring is in twee van de drie gevallen geen lichaam.

Voorbeeld 3.9 (UPC) Een eenvoudig voorbeeld van foutendetectie wordt gebruikt in de *streepjescode* die hoort bij de *Universal Product Code*, UPC. Deze codering, sinds 1973 in gebruik, is terug te vinden op heel veel consumptiegoederen, en bestaat uit 12 decimale cijfers, hier te representeren als een element

$v \in \mathbb{Z}/10\mathbb{Z}^{12}$. Het eerste cijfer geeft een indicatie van het type product, de volgende 5 cijfers geven de fabrikant aan, en daarna volgen 5 cijfers voor het product. Het laatste cijfer v_{12} is het zogenaamde *check*-cijfer en wordt geheel bepaald door de regel

$$\langle (3,1,3,1,3,1,3,1,3,1,3,1), v\rangle = 0 \in \mathbb{Z}/10\mathbb{Z},$$

dat wil zeggen,

$$3\sum_{i=1}^{6} v_{2i-1} + \sum_{i=1}^{6} v_{2i} = 0 \in \mathbb{Z}/10\mathbb{Z}.$$

Voorbeeld 3.10 Het *international standard book number* (ISBN) is een ander voorbeeld van een 'codering' die van een check-cijfer gebruik maakt. Het ISBN is (op te vatten als) een vector w uit $(\mathbb{Z}/11\mathbb{Z})^{10}$ met de restriktie dat de eerste 9 coördinaten ongelijk aan $\overline{10}$ zijn; ze worden elk weergegeven als decimale cijfers, en representeren land (1 tot 3 cijfers), uitgever (1 tot 5 cijfers), en boek. Het allerlaatste cijfer is weer een element dat als check-digit dient, en dat wél elk element uit $\mathbb{Z}/11\mathbb{Z}$ kan zijn, weergegeven door een decimaal (de kleinste niet-negatieve representant van de restklasse) of de letter X (die de restklasse $\overline{10}$ representeert). De waarde ervan wordt volledig bepaald door de regel

$$\langle (10,9,8,7,6,5,4,3,2,1), w\rangle = \sum_{i=1}^{10} (11-i)w_i = 0 \in \mathbb{Z}/11\mathbb{Z}.$$

Voorbeeld 3.11 Op de meeste producten wordt sinds 1 januari 2005 het *European Article Number* vermeld in plaats van de UPC. Dit EAN bestaat uit een vector x ter lengte 13 over $\mathbb{Z}/10\mathbb{Z}$, waar het laatste cijfer x_{13} een check-digit is, bepaald door

$$\langle (1,3,1,3,1,3,1,3,1,3,1,3,1), x\rangle = 3\sum_{i=1}^{6} v_{2i} + \sum_{i=1}^{7} v_{2i-1} = 0 \in \mathbb{Z}/10\mathbb{Z}.$$

Oude UPC aanduidingen kunnen simpel vervangen worden door een EAN door de oude vector vooraf te laten gaan door een 0. De nieuwe EAN aanduidingen geven land, fabrikant, en product met een wisselend aantal cijfers weer. Per 1 januari 2007 werden ook de oude ISBNumbers vervangen door een 13-cijferige ISBN-13, die net zo is opgebouwd als de EAN; de eerste 3 cijfers van de nieuwe aanduiding voor boeken zullen voorlopig altijd 978 zijn, en staan voor het universele 'Bookland'.

Stelling 3.12 *Een code met minimumafstand d is e-foutenverbeterend dan en slechts dan wanneer $d \geq 2e+1$.*

Bewijs. Veronderstel dat er een vector v bestaat op afstand kleiner of gelijk e van twee codewoorden w_1 en w_2 in een code met minimuafstand $d \geq 2e+1$. Dan is

$$h(w_1, w_2) \leq h(w_1, v) + h(v, w_2) \leq 2e,$$

terwijl voor de codewoorden $w_1 \neq w_2$

$$h(w_1, w_2) \geq d \geq 2e+1.$$

Dat is een tegenspraak, tenzij $w_1 = w_2$, en dus is er hoogstens één vector op afstand ten hoogste e als de minimumafstand ten minste $2e+1$ is, en de code is dus e-foutenverbeterend.

Veronderstel, voor de omkering, dat de minimumafstand d hoogstens $2e$ is, dan zijn er twee codewoorden u, w op afstand $d \leq 2e$ van elkaar. Vorm nu een rij x_i met

$$u = x_1, x_2, \ldots, x_d, x_{d+1} = w$$

van $d+1$ vectoren, waarbij x_{i+1} steeds op precies 1 coördinaat van x_i verschilt. Dan is er een vector v, namelijk $v == x_{\lfloor d/2 \rfloor}$, die op hoogstens e coördinaten van u en op niet meer dan e coördinaten van w verschilt. Maar dan zijn zowel u als w decoderingen van v op afstand hoogstens e: e fouten kunnen niet altijd uniek verbeterd worden, en de code is niet e-foutenverbeterend.

Opmerking 3.13 Een andere formulering van dezelfde stelling is: een code is e foutenverbeterend dan en slechts dan als alle bollen met straal e om verschillende codewoorden disjunct zijn. Hier moet je 'bol' wel met een korreltje zout nemen: een bol met straal r om een woord w in $\mathbb{F}_q^n$ bestaat uit alle vectoren met Hamming-afstand ten hoogste r tot w. Dat zijn er natuurlijk maar eindig veel, en dat zo'n verzameling een soort bolletje vormt komt door de eigenschap dat de Hamming-afstand net als de gewone afstand in de $\mathbb{R}^n$ een *metriek* vormt (zie de opgaven), inclusief een driehoeksongelijkheid (die ook in het bewijs gebruikt werd).

De centrale vraag voor de coderingstheorie is gelegen in de spanning tussen de twee eisen die we graag aan codes zouden opleggen. In de eerste plaats willen we zoveel mogelijk informatie oversturen, dat wil zeggen de rate k/n zo groot mogelijk maken (en de redundantie klein). Anderzijds willen we graag zoveel mogelijk fouten kunnen herstellen, dus e (en daarmee volgens Stelling 3.12 ook d) zo groot mogelijk kiezen; maar een grote minimumafstand betekent veel vectoren buiten de codewoorden, dus juist een lage rate.

Er bestaan heel veel grenzen op de omvang van goede codes. We geven hier een bekende bovengrens op het aantal codewoorden als de dimensie en het aantal fouten e dat verbeterd moet kunnen worden gegeven is.

Stelling 3.14 (Hamming grens) *Voor een code $\mathcal{C}$ over $\mathbb{F}_q$ van lengte n en minimumafstand groter of gelijk $2e+1$ geldt:*

$$\#\mathcal{C} \leq \frac{q^n}{\sum_{i=0}^{e} \binom{n}{i} (q-1)^i}$$

codewoorden.

Bewijs. Laat c een woord uit de code $\mathcal{C}$ in $\mathbb{F}_q^n$ zijn. De bollen met straal e rond c bevat in totaal

$$\sum_{i=0}^{e} \binom{n}{i} (q-1)^i$$

verschillende vectoren, want een woord op afstand i van codewoord c kan op $\binom{n}{i}$ plaatsen op $(q-1)^i$ manieren van c verschillen. Aan de ene kant moeten al deze bollen disjunct zijn (zie de voorgaande Opmerking), dus er zijn in totaal

$$\#\mathcal{C} \cdot \sum_{i=0}^{e} \binom{n}{i} (q-1)^i$$

verschillende vectoren in al die bollen. Dit aantal vectoren is begrens door het totaal aan vectoren in de hele ruimte, q^n: de ongelijkheid volgt.

Voorbeeld 3.15 Veronderstel dat we een binaire code willen maken van lengte $n = 5$, waarmee we 2 fouten kunnen verbeteren. De Hamming grens geeft dan aan dat de code niet meer dan $2^5/(1 + 5 + 10) = 2$ vectoren kan bevatten: alleen de repetitiecode voldoet! Inderdaad, volgens Stelling 3.12 is d minstens 5.

Opmerkingen 3.16 Veronderstel nu dat we een k-dimensionale code willen construeren in een n-dimensionale vectorruimte over $\mathbb{F}_q$ (met $k < n$). Het volstaat dan om k onafhankelijke vectoren te kiezen, en $\mathcal{C}$ te laten bestaan uit de $\mathbb{F}_q$-lineaire combinaties van deze k basisvectoren, die we soms schrijven als een $k \times n$ matrix G, de zogenaamde *generatormatrix*. Elk codewoord is dus een $\mathbb{F}_q$-lineaire combinatie van deze k rijvectoren.

Een andere manier om een k-dimensionale deelruimte van $\mathbb{F}_q^n$ te bepalen, is door $n - k$ onafhankelijke lineaire vergelijkingen op te schrijven en daarvan de oplossingsruimte te bepalen: de k-dimensionale ruimte is dan de kern van een $n \times (n-k)$ matrix H, die de *parity-check matrix* wordt genoemd, omdat deze precies alle lineaire condities op de codewoorden vastlegt. Hier schrijf je een woord w als een rijvector ter lengte n, en om na te gaan of het in de code zit vermenigvuldig je deze vector van *rechts* met de matrix H. Het woord zit in de code dan en slechts dan als $w \cdot H = 0$.

Het verband tussen de generatormatrix G en de parity-checkmatrix H wordt gegeven door de relatie $G \cdot H = 0$: immers, elke rij van G is een codewoord, en het product daarvan met H geeft de nulvector.

Voorbeeld 3.17 (Binaire Hamming code) Er bestaat een binaire Hamming code van lengte $2^r - 1$ en dimensie $2^r - 1 - r$ voor elke $r \geq 2$. We geven de constructie voor $r = 3$.

Constureer de code ter lengte $n = 2^3 - 1 = 7$ over $\mathbb{F}_2$ van dimensie $k = 4$ door de parity-check matrix H_3 op te schrijven: de rijen van H_3 bestaan uit alle mogelijke drietallen bits, met uitzondering van 3 nullen; een systematische manier om dat te doen is door de binaire schrijfwijze van alle getallen van 1 tot en met 7 te gebruiken:

$$H_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

(Merk op dat we hier, ongebruikelijk, het minst significante bit van een getal voorop schrijven; dat zorgt ervoor dat we makkelijker H_4 uit H_3 maken.) Laat deze 7×3 matrix de parity-checkmatrix van de code $\mathcal{C}_3$ over $\mathbb{F}_2$ zijn; de codewoorden zijn dus die rijvectoren w ter lengte 7 over $\mathbb{F}_2$ die $\langle w, k \rangle = 0$ hebben met elk van de drie kolommen van H. Daarvan zijn er 2^4, bijvoorbeeld alle lineaire combinaties van de rijen van

$$G_3 = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

over $\mathbb{F}_2$, die de kern van H_3 opspannen.

De Hammingcode is 1-foutenverbeterend voor alle $r \geq 2$.