Entropie en Huffman-codering

T. H. Koornwinder (thk@science.uva.nl)

Syllabus bestemd voor mastercourse Datacompressie, UvA, 23 januari 2004

Conventies Als we in het vervolg $\log x$ schrijven, dan bedoelen we ${}^2\log x$, dus de logaritme met grondtal 2. Met *desda* bedoelen we "dan en slechts dan als". Bij het samenstellen van deze syllabus is gebruik gemaakt van referenties [1], [2] en [3].

1 Het begrip informatie

Shannon [4] legde in 1948 de grondslagen van de informatietheorie en hij voerde ook het begrip entropie in verband met informatie in. Stel we hebben een eindige verzameling $S = \{s_1, s_2, \ldots, s_m\}$, waarin de elementen s_i mogelijke gebeurtenissen voorstellen die van elkaar onafhankelijk zijn. Bijvoorbeeld de verschillende karakters (letters, etc.) die in een tekst in een bepaalde taal (zoals Engels of Nederlands) kunnen voorkomen. Of een mogelijk weertype dat op 23 januari 2004 om 10 uur 's ochtends in Amsterdam kan voorkomen. We nemen aan dat gebeurtenis s_i waarschijnlijkheid p_i heeft. Dus $0 \le p_i \le 1$ en $\sum_{i=1}^m p_i = 1$.

Definitie 1.1. De *informatie* $I(s_i)$ geleverd door een gebeurtenis s_i is gelijk aan

$$I(s_i) = -\log p_i. \tag{1.1}$$

Omdat $0 \le p_i \le 1$ zal $0 \le I(s_i) \le \infty$, waarbij $I(s_i) = 0$ als $p_i = 1$ en $I(s_i) = \infty$ als $p_i = 0$. Ook is $I(s_i)$ een monotoon dalende functie van p_i. Als $p_i = \frac{1}{2}$ dan $I(p_i) = 1$.

Naarmate de kans kleiner is dat een gebeurtenis plaats heeft, zijn we meer verrast als hij plaats heeft en geeft hij ons meer informatie, en zijn we ook meer bereid om er plaats aan te geven in ons informatiekanaal of onze informatiedrager.

Dank zij de logaritme in de definitie is informatie een additieve grootheid. Stel bijvoorbeeld dat we twee keer achter elkaar een gebeurtenis uit de verzameling S hebben en dat die twee achtereenvolgende gebeurtenissen onafhankelijk van elkaar zijn. De verzameling van alle mogelijke twee achtereenvolgende gebeurtenissen wordt beschreven door $S^2 = S \times S = \{(s_i, s_j) \mid i, j = 1, \ldots, m\}$. De waarschijnlijkheid van de gebeurtenis (s_i, s_j) is $p_i p_j$ en $I(s_i, s_j) = -\log(p_i p_j) = -\log p_i - \log p_j = I(s_i) + I(s_j)$. Dus de informatie gegeven door eerst de gebeurtenis s_i en dan de gebeurtenis s_j is gelijk aan de som van de informaties gegeven door resp. s_i en s_j.

Informatie is dus een meetbare grootheid waarbij het zinvol is om van een eenheid van informatie te spreken. Deze eenheid noemen we *bit*. Een gebeurtenis met waarschijnlijkheid $\frac{1}{2}$ geeft $-\log \frac{1}{2} = 1$ bit informatie. Een gebeurtenis met waarschijnlijkheid 2^{-k} geeft $-\log 2^{-k} = k$ bits informatie. Dit is mooi in overeenstemming met het vertrouwde begrip van bit. Als het aantal mogelijke gebeurtenissen gelijk is aan 2^k, elk met kans 2^{-k}, dan kunnen we aan elke mogelijke gebeurtenis een verschillend rijtje van k nullen en enen toekennen. Er zijn dus precies k bits van het vertrouwde type nodig om ons de informatie te geven welke gebeurtenis heeft plaats gevonden.

2 Entropie

We blijven werken met de verzameling S en de kans p_i op een gebeurtenis s_i uit S. We kunnen dus spreken van een (eindige) kansruimte (S, P). Omdat de informatie I gedefinieerd is als een functie op S, is het zinvol om van de *verwachtingswaarde* van I te spreken.

Definitie 2.1. De *entropie* $H(S)$ van de kansruimte (S, P) is gelijk aan de verwachtingswaarde van de informatie I, i.e.,

$$H(S) = -\sum_{i=1}^{m} p_i \log p_i = \sum_{i=1}^{m} p_i \log p_i^{-1}. \tag{2.1}$$

Hier definiëren we $p \log p$ voor $p = 0$ door continuïteit, dus gelijk aan $\lim_{p \downarrow 0} p \log p = 0$.

Als we $H(S)$ beschouwen op een kansruimte met een vast aantal van m elementen, waarbij de kansen $p_1, \ldots, p_m$ nog kunnen variëren, dan is $H(S)$ in feite een functie $H(p_1, \ldots, p_m)$ van m variabelen, waarbij $p_1, \ldots, p_m$ beperkt zijn tot de volgende deelverzameling van $\mathbb{R}^m$:

$$\{(p_1, \ldots, p_m) \mid p_1 + \cdots + p_m = 1 \quad \text{en} \quad \forall i \; p_i \geq 0\} \qquad \text{(een simplex)}.$$

Als $m = 2$ dan is $H(p_1, p_2)$ alleen afhankelijk van $p_1 \in [0, 1]$ omdat $p_2 = 1 - p_1$. Dan geldt:

$$H(p, 1-p) = -p \log p - (1-p) \log(1-p), \tag{2.2}$$

$$\frac{d}{dp} H(p, 1-p) = \log(1-p) - \log p. \tag{2.3}$$

Bekijk de grafiek van $p \mapsto H(p, 1-p)$. De functie is symmetrisch rond $p = \frac{1}{2}$ en bij $p = \frac{1}{2}$ neemt hij ook zijn maximumwaarde 1 aan. Bij $p = 0$ en 1 is de functie 0. Hij stijgt van $p = 0$ tot $\frac{1}{2}$ (loopt verticaal weg van $p = 0$) en hij daalt van $p = \frac{1}{2}$ tot 1.

Het gedrag dat we in dit eenvoudige geval zien, keert terug bij willekeurige waarden van m:

Stelling 2.2. *Er geldt:*

$$0 \leq H(p_1, \ldots, p_m) \leq H(m^{-1}, \ldots, m^{-1}) = \log m. \tag{2.4}$$

$H(p_1, \ldots, p_m)$ bereikt zijn absolute maximum $\log m$ in precies één punt, nl. als alle p_i aan elkaar gelijk zijn, dus gelijk zijn aan m^{-1}.
$H(p_1, \ldots, p_m)$ bereikt zijn absolute minimum 0 desda $p_i = 1$ voor zekere i, en dus de p_j met $j \neq i$ gelijk aan 0 zijn.

De entropie van de verzameling S van gebeurtenissen is de gemiddelde hoeveelheid informatie die we krijgen bij een gebeurtenis. Als alle m mogelijke gebeurtenissen ongeveer gelijke kans hebben, dan krijgen we per gebeurtenis gemiddeld informatie die dicht bij $\log m$ bits ligt. Dus als $m = 2^k$ dan ontavngn we juist de k bits per gebeurtenis die we nodig hebben om ons logboek bij te houden zo dat elke mogelijke gebeurtenis door zijn eigen rijtje van k nullen en enen wordt voorgesteld. Het is paradoxaal dat we bij zo'n kansverdeling nooit echt verrast zullen worden, maar toch erg veel informatie per keer krijgen.

Als echter de kansen van de m gebeurtenissen flink uiteenlopen, dan zal de entropie veel lager zijn, dus de gemiddelde informatie per gebeurtenis veel lager. Weliswaar worden we nu soms echt verrast, maar de echte verrassingen kunnen niet vaak genoeg voorkomen om de gemiddelde

verrassing per gebeurtenis hoog te houden. Voor het logboek ruimen we nu voor de kansrijke gebeurtenissen korte rijtjes nullen en enen in en voor de zeldzame gebeurtenissen langere rijtjes. Het zal blijken dat dit zo slim gedaan kan worden dat de gemiddelde benodigde ruimte per gebeurtenis in ons logboek de hoogte van de entropie redelijk volgt.

3 Entropie als absolute ondergrens voor compressie

Voor de natuurlijke logaritme is er de standaardongelijkheid

$$\ln x \leq x - 1, \qquad (x > 0). \tag{3.1}$$

Er geldt gelijkheid desda $x = 1$. Voor de logarime met grondtal 2 kunnen we de ongelijkheid herschrijven als

$$\log x \leq (\log e)\,(x - 1), \qquad (x > 0). \tag{3.2}$$

Dit leidt tot interessante ongelijkheden voor de entropie.

Lemma 3.1. *Laat* $p_1, \ldots, p_m > 0$, $q_1, \ldots, q_m > 0$, $\sum_{i=1}^m p_i = 1$, $\sum_{i=1}^m q_i \leq 1$. *Dan*

$$-\sum_{i=1}^m p_i \log p_i \leq -\sum_{i=1}^m p_i \log q_i, \tag{3.3}$$

waarbij gelijkheid geldt desda $p_i = q_i$ *voor alle* i.

Bewijs We gaan bewijzen dat het rechter lid minus het linker lid ≥ 0 is.

$$\begin{aligned} -\sum_{i=1}^m p_i \log(q_i p_i^{-1}) &\geq -(\log e) \sum_{i=1}^m p_i (q_i p_i^{-1} - 1) \\ &= (\log e) \left(\sum_{i=1}^m p_i - \sum_{i=1}^m q_i \right) = (\log e) \left(1 - \sum_{i=1}^m q_i \right) \geq 0. \end{aligned}$$

Hier is (3.2) gebruikt. Uit de beschouwing wanneer (3.2) een gelijkheid is, zien we dat de eerste ongelijkheid in de afleiding hierboven een gelijkheid wordt desda $p_i = q_i$ voor alle i. Dan is vanzelf ook de som van de q_i's gelijk aan 1. □

Stel nu dat we een tekst aangeboden krijgen van lengte L, waarin een verzameling $S = \{s_1, \ldots, s_m\}$ van m karakters gebruikt wordt, en dat het aantal malen dat het karakter s_i in de tekst voorkomt, gelijk is aan Lp_i. Dus we kunnen $H(S) = -\sum_{i=1}^m p_i \log p_i$ als de bijbehorende entropie beschouwen.

Stel ook dat we een codering hebben van de karakters $s_1, \ldots, s_m$ waarbij s_i gecodeerd wordt door het binaire woord c_i. Hier is een *binair woord* een eindig en niet-leeg rijtje van nullen en enen. De *lengte* $l(c)$ van een binair woord c is het aantal nullen en enen in dat woord. Noem $C := \{c_1, \ldots, c_m\}$ de verzameling van de binaire codewoorden. De codewoorden hoeven niet allemaal even lang te zijn. Maar toch willen we dat de code *instantaan* is, d.w.z. dat we bij het inlezen van een rij nullen en enen gevormd uit achtereenvolgende codewoorden telkens met zekerheid weten wanneer we aan het eind zijn van een codewoord en met een volgend codewoord gaan beginnen.

Definitie 3.2. Zij C een eindige verzameling van binaire woorden. We noemen C *prefixvrij* als geen $u \in C$ een *prefix* is van een $w \in C$, d.w.z., als geen $w \in C$ te schrijven is als een $u \in C$ gevolgd door een binair woord v.

We zien nu onmiddellijk in dat een code C instantaan is desda C prefixvrij is.

Voorbeeld 3.3. De binaire woorden 0, 10, 110, 111 vormen een prefixvrije verzameling C. Als we een aantal binaire woorden uit C achter elkaar zetten, bijv.

10001111100010111

dan kunnen we moeiteloos en ondubbelzinnig de achtereenvolgende woorden uit C hrkennen:

10 0 0 111 110 0 0 10 111

Instantane codes hebben een andere belangrijke eigenschap:

Stelling 3.4. *(Ongelijkheid van Kraft)*
Zij $C = \{c_1, \ldots, c_m\}$ een instantane binaire code. Schrijf $l_i := l(c_i)$. Dan

$$\sum_{i=1}^{m} 2^{-l_i} \leq 1. \tag{3.4}$$

Bewijs De ongelijkheid geldt als de maximale woordlengte in C gelijk aan 1 is. Want dan $0 < 1$ als $m = 0$, $\frac{1}{2} < 1$ als $m = 1$ en $\frac{1}{2} + \frac{1}{2} = 1$ als $m = 2$. Stel nu dat de ongelijkheid geldt als de maximale woordlengte in C kleiner dan n is. Neem nu C met maximale woordlengte gelijk aan n. Bekijk C' gevormd door alle c in C te nemen met eerste bit 0, en dan die eerste bit weg te laten. Bekijk evenzo C'' gevormd door alle c in C te nemen met eerste bit 1, en dan die eerste bit weg te laten. Dan zijn C' en C'' prefixvrije codes met maximale lengte kleiner dan n, waarvoor dus Kraft's ongelijkheid geldt. Dan geldt voor C:

$$\sum_{c \in C} 2^{-l(c)} = \tfrac{1}{2} \sum_{c' \in C'} 2^{-l(c')} + \tfrac{1}{2} \sum_{c'' \in C'} 2^{-l(c'')} \leq \tfrac{1}{2} + \tfrac{1}{2} = 1.$$

Hiermee is de stelling bewezen op grond van volledige inductie naar de maximale woordlengte in C. □

Terug nu naar ongelijkheid (3.3), met p_i gelijk aan aantal voorkomens van karakter c_i in een tekst gedeeld door de lengte van de tekst. Zij $C = \{c_1, \ldots, c_m\}$ een instane binaire code voor de karakterverzameling $S = \{s_1, \ldots, s_m\}$ en neem q_i gelijk aan $2^{-l(c_i)}$. De ongelijkheid van Kraft geeft dan dat $\sum_{i=1}^{m} q_i \leq 1$, precies wat in (3.3) geëist wordt. Als we de aangeboden tekst coderen door S om te zetten in C, dan is het rechterlid van (3.3) gelijk aan $-\sum_{i=1}^{m} p_i \log q_i = \sum_{i=1}^{m} p_i\, l(c_i)$, dus aan de gemiddelde bitlengte van de gecodeerde karakters in de tekst. Dus we hebben bewezen:

Stelling 3.5. *Als de karakters van een tekst met een instantane code gecodeerd worden, dan is de gemiddelde bitlengte van de gecodeerde karakters in de tekst groter dan of gelijk aan de uit de karakterfrequenties in de tekst bepaalde entropie. Er geldt gelijkheid dan en slechts dan als voor elk karakter s_i met codewoord c_i geldt dat $2^{-l(c_i)}$ gelijk is aan het aantal voorkomens van s_i gedeeld door de lengte van de tekst.*

Stelling 3.4 (de ongelijkheid van Kraft) heeft een omgekeerde:

Stelling 3.6. *Als er een rijtje positieve gehele getallen*

$$l_1 \leq l_2 \leq \ldots \leq l_m$$

gegeven is zo dat de ongelijkheid (3.4) *geldt, dan bestaat er een instantane binaire code* $C = \{c_1, \ldots, c_m\}$ *met* $l(c_i) = l_i$.

Om dit in te zien, en ook voor later gebruik, introduceren we nu binaire bomen.

Definitie 3.7. Een *binaire boom* is een speciaal soort gerichte graaf. Hij bestaat, zoals elke gerichte graaf, uit punten en gerichte lijnen. De volgende eigenschappen karakteriseren een binaire boom:

- Er is één speciaal punt, de *wortel*, die we bovenaan tekenen. Bij de wortel komt gen enkele lijn aan.
- Elke gerichte lijn loopt naar linksonder of naar rechtsonder. Omdat de lijnen dus altijd naar beneden gaan, hoeven we de lijnen niet als pijlen te tekenen.
- Bij elk punt ongelijk aan de wortel komt precies één lijn aan.
- Vanuit elk punt vertrekken 2, 1, of 0 lijnen.
- Elk punt is vanuit de wortel te bereiken.

In een binaire boom noemen we de lijnen *takken*. We noemen de punten vanwaar geen takken vertrekken *bladeren*. We noemen een punt b een *kind* van een punt a als er een tak van a naar b loopt. Dan noemen we a *ouder* van b. Als twee punten b en c kinderen zijn van dezelfde ouder, dan noemen we b en c *broers* (of *zusters*) van elkaar. Een tak die naar linksonder loopt, coderen we met 0, en een tak die naar rechtsonder loopt met 1. Elk punt kan op een unieke manier via een aantal opeenvolgende takken bereikt worden. Deze opeenvolgende takken leveren een rijtje nullen en enen, een *binair woord*, dat het betreffende punt uniek bepaalt. De *lengte* van dit woord komt overeen met het *niveau* van het punt in de boom. Hier moeten we ook het *lege binaire woord* meenemen, dat overeenkomt met de wortel.

Met elke prefixvrije verzameling C van binaire woorden kunnen we een unieke *binaire boom* associëren zo dat de binaire woorden uit C precies de bladeren (eindpunten) van de boom zijn. Omgekeerd geeft elke binaire boom een unieke prefixvrije verzameling C bestaande uit de binaire woorden voor de bladeren van de boom. (Stilzwijgend nemen we hierbij aan dat C niet leeg is en dat de boom niet de wortel als enige punt heeft.) In Voorbeeld 3.3 heeft de wortel twee kinderen 0 en 1, waarvan 0 een blad is en 1 twee kinderen 10 en 11 heeft. 10 is een blad en 11 heeft twee kinderen 110 en 111, die beide bladeren zijn.

Bewijs van Stelling 3.6 Als de maximale lengte l_m gelijk is aan 1, dan is, gezien de ongelijkheid (3.4), $m = 1$, $l_1 = \frac{1}{2}$ of $m = 2$, $l_1 = l_2 = \frac{1}{2}$. We realiseren dit met $c_1 = 0$ en (als $m = 2$) $c_2 = 1$.

Stel nu dat we voor elke gegeven $l_1 \leq \ldots \leq l_m$ met $l_m < L$ al een prefixvrije code gerealiseerd hebben. Laat nu $l_m = L$. Het aantal i's met $l_i = l_m = L$ kan even of oneven zijn. Er volgt uit (3.4) dat, als het aantal oneven is, dan $\sum_{i=1}^{m} 2^{-l_i} \leq 1 - 2^{-L}$. Dus dan kunnen we een $l_{m+1} = L$ toevoegen met behoud van ongelijkheid (3.4). Als we dan $l_1 \leq \ldots \leq l_{m+1}$ kunnen realiseren

met een prefixvrije code, dan kunnen we dit ook door weglating van c_{m+1} voor $l_1 \leq \ldots \leq l_m$. We mogen dus aannemen dat het aantal i's met $l_i = l_m = L$ even is, zeg $2k$. We vervangen de $2k$ l_i's gelijk aan L door k l_i's gelijk aan $L-1$. Voor de nieuwe rij van $m-k$ l_i's geldt de ongelijkheid (3.4), dus uit de hypothese volgt dat deze gerealiseerd kunnen worden door een prefixvrije code $\{c_1, \ldots, c_{m-k}\}$. Hang nu aan elk van de k bladeren $c_{m-2k+1}, \ldots, c_{m-k}$ in de bijbehorende binaire boom (alle op niveau $L-1$) twee kinderen, en vervang $c_{m-2k+1}, \ldots, c_{m-k}$ door de zo verkregen $2k$ bladeren in de boom op niveau L. Dit geeft de gevraagde realisatie door een prefixvrije binaire code.

Met volledige inductie naar L hebben we zo de stelling bewezen. □

Opmerking 3.8. Met essentieel hetzelfde bewijs als hierboven zien we:
Als er een rijtje positieve gehele getallen $l_1 \leq l_2 \leq \ldots \leq l_m$ gegeven is zo dat de gelijkheid

$$\sum_{i=1}^{m} 2^{-l_i} = 1 \tag{3.5}$$

geldt, dan bestaat er een instantane binaire code $C = \{c_1, \ldots, c_m\}$ met $l(c_i) = l_i$ en zo dat in de bijbehorende binaire boom elke ouder twee kinderen heeft.

Omgekeerd, als $C = \{c_1, \ldots, c_m\}$ een prefixvrije binaire code is met $l(c_i) = l_i$ en zo dat in de bijbehorende binaire boom elke ouder twee kinderen heeft, dan zal gelijkheid (3.5) gelden.

4 Shannon-Fano code

Laten we recapituleren waar het ons om ging. We hebben een eindige kansruimte $S = \{s_1, \ldots, s_m\}$ met bijbehorende kansen p_i. We zoeken naar een instantane binaire code $C = \{c_1, \ldots, c_m\}$ met $l_i := l(c_i)$ (of equivalent naar een rijtje $l_1 \leq l_2 \ldots \leq l_m$ dat voldoet aan ongelijkheid (3.4)) zo dat $\sum_{i=1}^{l} p_i l_i$ zo klein moeglijk wordt. We weten al dat de entropie een absolute ondergrens is:

$$H(S) = -\sum_{i=1}^{m} p_i \log p_i \leq \sum_{i=1}^{l} p_i l_i. \tag{4.1}$$

Ook weten we dat de ondergrens bereikt (en met een instantane binaire code gerealiseerd) kan worden desda $p_i = 2^{-l_i}$ voor alle i.

We kunnen direct al goedkope winst maken in (4.1). Als we een instantane binaire code hebben zo dat er in de bijbehorende binaire boom ouders met 1 kind voorkomen, dan kunnen we die punten in de boom weghalen. Hierdoor zijn er uiteindelijk alleen nog maar ouders met 2 kinderen en zijn sommige bladeren omhoog gekomen. Het effect is dat in sommige binaire codewoorden er een of meer bitplaatsen zijn verdwenen, waardoor de lengte van zo'n woord kleiner is geworden. Hierdoor wordt het rechterlid van (4.1) kleiner: de code is efficiënter geworden.

Als niet alle p_i's negatieve machten van 2 zijn, dan kunnen we wel l_i's nemen zo dat 2^{-l_i} zo dicht mogelijk onder p_i ligt. Dus

$$l_i := \lceil \log p_i^{-1} \rceil, \tag{4.2}$$

dit is het kleinste gehele getal $\geq \log p_i^{-1}$. Met deze keuze van l_i's zal ongelijkheid (3.4) gelden. Dus we kunnen een instantane binaire code C voor de l_i's realiseren. Deze code heet de *Shannon-Fano code*.

Stelling 4.1. *(Shannon's Noiseless Coding Theorem) Voor de Shannon-Fano code geldt:*

$$H(S) = -\sum_{i=1}^{m} p_i \log p_i \leq \sum_{i=1}^{l} p_i l_i < H(S) + 1. \tag{4.3}$$

Bewijs Uit (4.2) volgt dat $l_i - 1 < -\log p_i < l_i$. Vermenigvuldig alles met p_i, sommeer alles over i en gebruik dat $\sum_{i=1}^{m} p_i = 1$. Dit levert het gewenste resultaat. □

We kunnen dus zo coderen dat de gemiddelde bitlengte per karakter de entropie met minder dan 1 overschrijdt. Bovendien kunnen we de Shannon-Fano code nog verbeteren door de een-kinder ouders in de binaire boom weg te halen. Hoeveel beter het daardoor wordt, zal er van afhangen hoe we precies de binaire code vaststellen behorend bij gegeven lengtes l_i die aan ongelijkheid (3.4) voldoen. Het recursieve voorschrift in het bewijs van Stelling 3.6 zouden we eigenlijk preciezer moeten maken. In plaats daarvan zullen we kijken naar de code die het rechterlid in (4.1) zeker minimaliseert: de Huffman-code.

5 Huffman-codering

Normaal worden karakters gecodeerd met een vast aantal bits, bijv. de 7 bits van de ascii-code of de 8 bits van de uitgebreide ascii-code. Dit heeft het voordeel dat men weet na hoeveel bits de code voor het volgende karakter begint, maar het nadeel dat men evenveel bits voor een veel voorkomend als een weinig voorkomend karakter moet gebruiken. Als men karakters codeert in verschillende bitlengtes, en men werkt alleen met nullen en enen, dan moeten er eisen aan de code gesteld worden opdat de ontvanger altijd zeker weet, wanneer een binnenkomende rij nullen en enen de informatie voor een karakter voltooid heeft en aan de informatie voor een volgend karakter begint. Dit wordt gerealiseerd desda de verzameling codewoorden de eigenschap van prefixvrijheid heeft.

Stel nu dat een eindige verzameling S gegeven is en dat met elke $s \in S$ een positief reëel getal $f(s)$ (een *gewicht*) geassocieerd is. Denk bij S aan de gebruikte karakters in een gegeven tekst, en bij $f(s)$ aan het aantal keren dat s in de tekst voorkomt, of het aantal keren dat s voorkomt gedeeld door de lengte van de tekst. We zoeken nu een een bijectieve afbeelding $\Phi\colon S \to C$ van S naar een prefixvrije verzameling C van binaire woorden zo dat $\sum_{s\in S} f(s)l(\Phi(s))$ zo klein mogelijk is.

Equivalente formulering Gegeven is een niet-dalende rij van m positieve reële getallen

$f_1 \leq f_2 \leq \ldots \leq f_m$.

Vind een prefixvrije verzameling C van m binaire woorden $c_1, c_2, \ldots, c_m$ zo dat

$$F(C) := \sum_{i=1}^{m} f_i\, l(c_i)$$

minimaal is.

Het is eenvoudig in te zien dat de boom voor een C met minimale $F(C)$ in ieder geval de volgende eigenschappen moet hebben (omdat we anders een boom kunnen maken met kleinere $F(C)$):

- Er zijn geen punten vanwaar slechts 1 tak vertrekt.
- Er zijn geen i en j zo dat $f_i < f_j$ en $l(c_i) < l(c_j)$.

Dan zien we in dat in een boom met minimale $F(C)$ moet gelden dat $l(c_1) = l(c_2)$ en dat, na eventuele herordening van de c_i's met maximale lengte, c_1 en c_2 dezelfde ouder hebben.

Het algoritme van Huffman construeert recursief een C met minimale $F(C)$:

Beginstap $m = 2$. Neem $c_1 = 0$, $c_2 = 1$.

Recursieve stap Stel dat we voor de rij van $m-1$ getallen $f_1 + f_2, f_3, \ldots, f_m$ een bijpassende verzameling C', dus ook een binaire boom B', met de minimaliteitseigenschap gevonden hebben. Neem in B' een blad met gewicht $f_1 + f_2$. Laat uit dit blad twee takken vertrekken uitkomend in 2 bladeren met gewichten f_1 en f_2. De zo uit B' geconstrueerde boom B levert C als gevraagd.

Er kan bewezen worden (niet erg moeilijk) dat dit algoritme inderdaad een C oplevert met $F(C)$ minimaal. In bovenstaande beschrijving van het algoritme zijn nog keuzevrijheden. Namelijk, als in de recursieve stap er al andere f_i's zijn gelijk aan $f_1 + f_2$, dan zijn er verschillende mogelijkheden om f_1 en f_2 als bladeren aan een f_i in B' te hangen. Elke keuze zal echter tot dezelfde minimale $F(C)$ leiden. Ook zal, vanwege de minimaliteitseigenschap de l_i's in de gevonden boom moeten voldoen aan de eis $l_i > l_j \Rightarrow f_i \leq f_j$.

Voorbeeld 5.1. Gegeven is een gewichtenrij

$\underline{3}$, $\underline{4}$, 4, 6, 7, 15

De twee kleinste gewichten zijn onderstreept. We vervangen deze door hun som $3 + 4$, dus door 7, waarboven we een streep zetten, we brengen de nieuwe rij gewichten weer op volgorde, en we onderstrepen weer de twee eerste (kleinste) gewichten in de rij:

$\underline{4}$, $\underline{6}$, $\overline{7}$, 7, 15

Zo gaan we door:

$\underline{7}$, $\underline{7}$, $\overline{10}$, 15

$\underline{10}$, $\overline{\underline{14}}$, 15

$\underline{15}$, $\overline{\underline{24}}$

$\overline{39}$

Nu lopen we in omgekeerde volgorde langs de recursie om de boom te tekenen. De laatste regel geeft een boom met alleen een wortel met gewicht 39. Daar hangen we twee bladeren aan met gewichten 15 en 24. Aan het blad met bovenstreept gewicht 24 hangen we twee bladeren met gewichten die onderstreept zijn in de regel erboven, dus 10 en 14. Enzovoorts.

Hoe goed is de Huffman-codering, als we deze vergelijken met de entropie? Het antwoord wordt gegeven door (4.3). De gemiddelde bitlengte per karakter bij Huffman-codering is $\geq H(S)$ en $< H(S) + 1$, maar vanwege de minimaliteit ook $\leq$ de gemiddelde bitlengte per karakter bij Shannon-Fano-codering.

Uiteraard moet de (de)coderingstabel ook in de gecodeerde file worden meegegeven. Dit neemt extra ruimte in beslag, die relatief groot kan zijn als C groot is. Maar als C uit hoogstens 128 of 256 elementen bestaat, dan is de decoderingstabel bij een file van enige omvang maar een kleine overhead.

Een nadeel van Huffman-codering is dat we de file twee keer moeten doorlopen: eenmaal om de frequenties van de karakters te tellen en eenmaal om de codering uit te voeren. Om hieraan tegemoet te komen is de *adaptieve Huffman-codering* opgezet. Hier begin je met toekennen van gewicht 1 aan alle mogelijk voorkomende karakters en zet je de bijbehorende Huffman-boom op. Daarna ga je karakters inlezen. Na elk ingelezen karakter wordt het gewicht van dat karakter

opgehoogd, wat mogelijk aanleiding geeft tot bijstellen van de Huffman-boom. Bij het decoderen moet deze boom evenzo worden bijgehouden om te weten wat de actuele coderingstabel is. Dit lijkt zeer tijdrovend, maar het kan slim en relatief snel gebeuren. Hieraan zijn de namen van Faller, Gallager en Knuth verbonden. Men spreekt ook wel van het *FGK-algoritme*.

Er is een probleem met de adaptieve Huffman-codering als een beperkt aantal karakters een tijd lang vaak zijn voorgekomen, maar dan plotseling niet of weinig, terwijl een ander karakter vaak gaat voorkomen. Dan duurt het lang totdat dit nieuwe karakter voldoende gewicht heeft opgebouwd om met weinig bits gecodeerd te worden. Een remedie werd gesuggereerd door Gallager. Vermenigvuldig periodiek, na inlezen van een vast aantal karakters, alle gewichten van dat moment met een getal < 1, bijv. $\frac{1}{16}$. Hierdoor krijgen recente trendwijzigingen in de file meer nadruk dan wat langer geleden gebeurde. Als er echter geen trendwijziging is, dan blijft de vermenigvuldiging van de gewichten zonder effect, wat ook de bedoeling is.

6 Entropie van hogere orde

Deze paragraaf geeft een uitbreiding van wat we in §2 over entropie bespraken. Stel nu dat we twee kansruimtes (S, P) en (T, Q) hebben, waarbij $T = \{t_1, \ldots, t_n\}$ met kans q_j voor de gebeurtenis t_j. Met de twee kansruimtes zijn entropiewaarden $H(S)$ en $H(T)$ geassocieerd. Er zijn diverse samenstellingen van S en T mogelijk waarmee een nieuwe entropie geassocieerd kan worden.

1. *Direct product* De kansruimte $(S \times T, P \times Q)$ bestaat uit de gebeurtenissen (s_i, t_j) met kans $p_i q_j$, $i = 1, \ldots, m$, $j = 1, \ldots, n$ (dus de gebeurtenissen s_i en t_j zijn onafhankelijk van elkaar). We rekenen eenvoudig na dat

$$H(S \times T) = H(S) + H(T). \tag{6.1}$$

   We kunnen ons voorstellen dat we eerst een bericht binnenkrijgen van type S, daarna van type T. De gemiddelde informatie die we na die twee berichten hebben binnengekregen is de som van de gemiddelde informatie bij het eerste bericht en bij het tweede bericht. Denk bijvoorbeeld aan het na elkaar ontvangen van twee karakters die onafhankelijk van elkaar zijn. Hier is $T = S$, $Q = P$.

2. *Fusie (amalgamation)* De elementen s_i van S worden samengevoegd in groepjes t_j, die de elementen van T vormen. De kans q_j voor t_j is de som van de kansen p_i van de s_i die samen t_j vormen. Dan

$$H(S) \geq H(T). \tag{6.2}$$

   Door de fusie daalt de gemiddelde informatie per gebeurtenis. Bijvoorbeeld, in plaats van een letter door te krijgen, wordt ons alleen bericht of het een klinker of medeklinker is. Dan daalt de entropie. Het bewijs van (6.2) volgt uit (3.3) (met de p_i's en q_i's verwisseld).

3. *Samenvoeging (join)* Een ruimte $S \wedge T$ met elementen (s_i, t_j) met kans r_{ij}, $i = 1, \ldots, m$, $j = 1, \ldots, n$ zo dat $p_i = \sum_j r_{ij}$ en $q_j = \sum_i r_{ij}$. Dus S en T zijn allebei als fusies van $S \wedge T$ op te vatten. Dan kan bewezen worden dat

$$\max(H(S), H(T) \leq H(S \wedge T) \leq H(S) + H(T). \tag{6.3}$$

De eerste ongelijkheid in (6.3) volgt uit (6.2), de tweede uit (3.3).
Een direct product $S \times T$ is een speciaal geval van samenvoeging $S \wedge T$. Er geldt $H(S \wedge T) = H(S) + H(T)$ desda $S \wedge T = S \times T$. Denk als voorbeeld van samenvoeging aan het na elkaar ontvangen van twee karakters die niet persé onafhankelijk van elkaar zijn, zoals gebruikelijk in letters van woorden uit een bepaalde taal. Hier is $T = S$, $Q = P$. Algemener kunnen we naar de samenvoeging $\bigwedge^k S = S \wedge S \wedge \ldots \wedge S$ van k ruimtes S kijken. Dan

$$H\left(\bigwedge^k S\right) \leq H\left(\bigwedge^{k-1} S\right) + H(S) \leq kH(S). \tag{6.4}$$

Dus voor de entropie per gebeurtenis geldt dat $k^{-1}H\left(\bigwedge^k S\right) \leq H(S)$.

4. *Voorwaardelijke entropie* Neem S en T zoals in de definitie van $S \wedge T$. Dan is de voorwaardelijke entropie van S onder T gedefinieerd door

$$H(T \mid S) = -\sum_{i=1}^{m}\sum_{j=1}^{n} r_{ij} \log(r_{ij}/p_i). \tag{6.5}$$

We kunnen r_{ij}/p_i als de voorwaardelijke kans op gebeurtenis (s_i, t_j) opvatten onder gegeven dat gebeurtenis s_i heeft plaats gevonden. Dus $-\log(r_{ij}/p_i)$ is de informatie die gegeven wordt door gebeurtenis t_j als gegeven is dat daarvoor gebeurtenis s_i heeft plaats gevonden. Dan is $H(T \mid S)$ de verwachtingswaarde van deze voorwaardelijke informaties. Er kan eenvoudig worden nagerekend dat

$$\begin{aligned} H(T \mid S) &= H(S \wedge T) - H(S), & (6.6)\\ H(T \mid S) &\leq H(T). & (6.7) \end{aligned}$$

Gelijkheid in (6.7) geldt desda de gebeurtenissen uit T onafhankelijk zijn van de gebeurtenissen uit S.

Een ander belangrijk resultaat is het volgende.

Stelling 6.1. *Laat de kansruimte $(\widehat{S}, \widehat{P})$ een fusie zijn van de kansruimte (S, P) en zij (T, Q) ook een kansruimte. Dan*

$$H(T \mid S) \leq H(T \mid \widehat{S}). \tag{6.8}$$

Het bewijs kan teruggevoerd worden tot (3.1). We zijn speciaal geïnteresseerd in het volgende speciale geval van (6.8), (6.6) en (6.4):

$$H\left(S \mid \bigwedge^k S\right) \leq H\left(S \mid \bigwedge^{k-1} S\right) \leq \ldots \leq H\left(S \mid S\right) \leq H(S), \tag{6.9}$$

$$H\left(\bigwedge^{k+1} S\right) = H\left(\bigwedge^k S\right) + H\left(S \mid \bigwedge^k S\right) = H(S) + \sum_{l=1}^{k} H\left(S \mid \bigwedge^l S\right), \tag{6.10}$$

$$(k+1)H(S) \geq H\left(\bigwedge^{k+1} S\right) \geq (k+1)\, H\left(S \mid \bigwedge^k S\right). \tag{6.11}$$

De voorwaardelijke entropie

$$H^{(k)}(S) := H\left(S \mid \bigwedge^k S\right) \tag{6.12}$$

wordt de *k-de orde entropie* van S genoemd. Als we S opvatten als de verzameling van mogelijke karakters, dan is $H^{(k)}(S)$ de voorwaardelijke entropie van het ontvangen van een karakter als de voorgaande k karakters bekend zijn. We kunnen voor de kansverdelingen in S, $S \wedge S$, etc. uitgaan van a priori aangenomen kansverdelingen, maar we kunnen ook benaderingen van deze (hypothetische) kansverdelingen vinden door frequenties van karakters in de gegeven tekst na te gaan, en ook de frequenties van de strings van lengte 2, 3 etc. Als onze tekst lengte L heeft en als het aantal keren dat het karakter s_i voorkomt gelijk is aan l_i, dan kennen we aan het karakter s_i kans $p_i = l_i/L$ toe. Evenzo, als de string $s_i s_j$ l_{ij} keer voorkomt, dan kennen we aan deze string kans $r_{ij} = l_{ij}/(L-1)$ toe. Voorts kennen we aan het karakter s_j voorwaardelijke kans l_{ij}/l_i toe om na een karakter s_i te komen.

In [1, Ch. 4] wordt een grote verzameling van Engelse teksten besproken, het *Brown corpus*, aan de hand waarvan veel taalstatistiek is bedreven. Het bestand bevat de meest uiteenlopende Engelse teksten, totaal ongeveer een miljoen woorden. In Engelse teksten wordt met 94 verschillende karakters gewerkt (kleine letters, hoofdletters, de spatie, leestekens, ...). De meest voorkomende karakters in het Brown corpus zijn (achter elk karakter wordt de kans gegeven):

spatie (0.174), e (0.098), i (0.070), a (0.061), o (0.059), i (0.055), n (0.055), s (0.050), r (0.047), h (0.041), l (0.032), d (0.030), c (0.023), u (0.021), m (0.019), f (0.018), p (0.015), g (0.015), w (0.014), y (0.013), b (0.011), komma (0.01), punt (0.08), v (0.08), k (0.05), T (0.03), ...

Dit levert een entropie van $H(S) = 4.47$ bits. Hier is S de verzameling van de 94 karakters. (Als alle karakters even waarschijnlijk zouden zijn, dan was de entropie 6.55 bits.)

De meest voorkomende strings van 2 karakters (*digrams*) in het Brown corpus zijn:

`'e '` (0.03), `' t'` (0.024), `'th'` (0.020), `'he'` (2.0), ...

Dit levert een eerste orde entropie $H^{(1)}(S) = 3.59$ bits (wat in overeenstemming is met ongelijkheid (6.7)). Voor de strings van 3 resp. 4 karakters in het Brown corpus worden entropiewaarden gevonden van $H^{(2)}(S) = 2.92$ resp. $H^{(3)}(S) = 2.33$ bits.

Een andere benadering, ook beschreven in [1, Ch. 4], probeert de entropie van het Engels te bepalen door aan proefpersonen teksten aan te bieden, waarbij men, na een beginstukje van de tekst, telkens het volgende karakter moet voorspellen. Als men niet het goede karakter zegt, dan mag men een nieuwe poging doen, net zo lang tot men het goed heeft. Hieruit krijgt de onderzoeker een idee van de waarde van $H^{(k)}(S)$ voor grote k, waarbij S de verzameling van karakters is, bij dit experiment beperkt tot de 26 letters en de spatie. De rij van deze k-de orde entropieën ($k = 0, 1, 2, \ldots$) is niet-stijgend dus nadert van boven een limiet, waarvan we veronderstellen dat hij positief is. Men veronderstelt, mede op grond van de experimenten, dat de limiet-entropie ligt tussen 0.6 en 1.3 bits.

Referenties

[1] T. C. Bell, J. G. Gleary and I. H. Witten, *Text compression*, Prentice Hall, 1990.

[2] R.W. Hamming, *Coding and information theory*, Prentice-Hall, 1980.

[3] G. A. Harris, P. D. Johnson and D. R. Hankerson, *Introduction to information theory and data compression*, Chapman & Hall, 2nd edition, 2003 (or 1st edition, 1998).

[4] `http://www-groups.dcs.st-and.ac.uk/~history/Mathematicians/Shannon.html`