Werkblad week 1

d.r.m.langers@pl.hanze.nl

20 oktober 2021

1 Informatie entropie

In de informatica wordt het begrip *entropie* (of *Shannon entropie*) gebruikt om kwantitatieve hoeveelheden informatie te meten. De entropie is nauw gerelateerd aan "de mate van verrassing" of "de mate van onzekerheid" dat bepaalde uitkomsten optreden. Een uitkomst die volkomen zeker is, zoals bijvoorbeeld het antwoord "ja" op de vraag "kwam de zon vanmorgen op?", is totaal niet verrassend en heeft een hele lage informatie-inhoud. Immers, je leert eigenlijk niets nieuws bij als je het antwoord "ja" krijgt, want je wist al van tevoren wel dat dat zo zou zijn. Een uitkomst die volkomen onzeker is, zoals bijvoorbeeld het antwoord "ja" op de vraag "is de pasgeboren baby een meisje?", die heeft daarentegen een grote informatie-inhoud en zou je wel verrassend kunnen noemen. In dit geval was de uitkomst vooraf letterlijk vrijwel 50/50, dus je was volkomen onzeker, maar de informatie in het antwoord "ja" heeft die onzekerheid opgeheven.

Opgave 1. Voor elk van de volgende paren van uitkomsten afzonderlijk, wat geeft je meer informatie, denk je? (Hint: de mediaan van de leeftijd van de Nederlandse bevolking is ongeveer 42.6 jaar; je mag voor het gemak aannemen dat de oogkleur bruin dominant is over de recessieve oogkleur blauw/groen.)

1. Het antwoord "ja" op de vraag "ben je ouder dan 42 jaar?"; of het antwoord "ja" op de vraag "ben je ouder dan 100 jaar?"

2. Het antwoord "ja" op de vraag "is de kleur van je ogen bruin?"; of, het antwoord "nee" op de vraag "is de kleur van je ogen bruin?"

3. Het antwoord "kop" op de vraag "wat was de uitkomst van de worp van een eerlijke munt?"; of het antwoord "6" op de vraag "wat was de uitkomst van de worp van een eerlijke dobbelsteen?"

De entropie $H(U)$ van een uitkomst U is nauw gerelateerd aan de kans $P(U)$ dat die uitkomst optreedt. In de voorbeelden hiervoor was de kans op de eerste uitkomst U_{zon} = "ja, de zon kwam vanmorgen op" gelijk aan $P(U_{\text{zon}}) = 1$ (oftewel 100%), en was de kans op de tweede uitkomst U_{baby} = "ja, de pasgeboren baby is een meisje" gelijk aan $P(U_{\text{baby}}) = \frac{1}{2}$ (of 50%). In het eerste geval was de informatie die je krijgt uit de uitkomst

gelijk aan nul, omdat je niks bijleerde van het antwoord. In formulevorm: $H(U_{\text{zon}}) = 0$. De informatie van de tweede uitkomst is echter positief en ongelijk aan nul, dus $H(U_{\text{baby}}) > 0$, omdat je er wel iets wijzer van werd.

Opgave 2. Voor elk van de volgende paren van uitkomsten afzonderlijk, schat of bereken de kans dat elk van die beide uitkomsten optreedt.

1. De kans op het antwoord "ja" op de vraag "ben je ouder dan 42 jaar?" en de kans op het antwoord "ja" op de vraag "ben je ouder dan 100 jaar?"

2. De kans op het antwoord "ja" op de vraag "is de kleur van je ogen bruin?" en de kans op het antwoord "nee" op de vraag "is de kleur van je ogen bruin?"

3. De kans op het antwoord "kop" op de vraag "wat was de uitkomst van de worp van een eerlijke munt?" en de kans op het antwoord "6" op de vraag "wat was de uitkomst van de worp van een eerlijke dobbelsteen?"

Als je de kansen op bepaalde uitkomsten weet, kun je hieruit ook de grootte van de hoeveelheid informatie die erbij hoort berekenen.

Hoeveelheden ongerelateerde informatie kun je optellen, oftewel de entropie van twee afzonderlijke uitkomsten U_1 en U_2 is per definitie gelijk aan de som van de entropieën van de twee uitkomsten apart. In formulevorm: $H(U_1, U_2) = H(U_1) + H(U_2)$. Voor kansen geldt ook zoiets: voor de kansen op twee onafhankelijke uitkomsten geldt echter dat $P(U_1, U_2) = P(U_1) \cdot P(U_2)$. Deze rekenregel ken je uit de statistiek.

Kortom, kansen op uitkomsten mag je vermenigvuldigen, maar hoeveelheden informatie van uitkomsten mag je optellen.

Opgave 3. Als je tegelijkertijd een eerlijke munt en een eerlijke dobbelsteen gooit, hoe groot is dan de kans dat de uitkomst een combinatie van "kop" en "6" is?

Opgave 4. De kans dat een willekeurige Nederlander een man is, is afgerond $P\,(\text{man}) = 0.50$. De kans dat een willekeurige Nederlander kleurenblind is, is afgerond $P\,(\text{kleurenblind}) = 0.05$. Toch blijkt de kans dat een willekeurige Nederlander een kleurenblinde man is niet gelijk te zijn aan $0.50 \cdot 0.05 = 0.025$. Deze kans blijkt aanzienlijk groter te zijn, namelijk $P\,(\text{kleurenblinde man}) = 0.042$. Waarom gaat de regel $P(U_1, U_2) = P(U_1) \cdot P(U_2)$ hier duidelijk niet op?

Vermenigvuldigingen kun je "omzetten" in optellingen door de logaritme te nemen: hopelijk weet je nog dat $\log(xy) = \log(x) + \log(y)$. Dit suggereert dat we de informatie van een gebeurtenis kunnen meten door de logaritme te nemen van de bijbehorende kans. Dit blijkt inderdaad zo te zijn.

Als de logaritme wordt genomen met grondtal 2, wat we in dit werkblad noteren als $\lg(x) = \log_2(x)$, dan wordt de uitkomst uitgedrukt in *bits* ("binary units"). We gebruiken de definitie

$$H(U) = -\lg(P(U))$$

Het min-teken is nodig omdat $P(U) \leq 1$, zodat $\lg(P(U)) \leq 0$. We willen echter juist dat $H(U) \geq 0$, want hoeveelheden informatie kunnen nooit negatief zijn!

Opgave 5. Laat zien dat de hoeveelheid informatie in het antwoord "kop" op de vraag "wat was de uitkomst van de worp van een eerlijke munt?" precies gelijk is aan 1 bit. Komt dit overeen met jouw eigen beeld van wat een "bit" is?

Opgave 6. Bereken de hoeveelheid informatie in het antwoord "6" op de vraag "wat was de uitkomst van de worp van een eerlijke dobbelsteen?".

Het is overigens ook mogelijk om een logaritme te gebruiken met grondtal 10: $H(U) = -\log(P(U))$, met als eenheid "digital units" genaamd *dits*. Of grondtal $e = 2.718\ldots$: $H(G) = -\ln(P(G))$, in eenheden van "natural units" genaamd *nats*. Deze zijn direct in elkaar om te rekenen met de gelijkheid $\log_b(x) = \frac{\log(x)}{\log(b)}$.

Opgave 7. Laat zien dat een "natural unit" overeenkomt met net iets minder dan anderhalve bit, en dat een "digital unit" overeenkomt met net iets minder dan $3\frac{1}{3}$ bits.

We passen dit idee nu eens toe op een dichotoom attribuut dat twee verschillende waarden kan aannemen. Als de kans op de ene waarde gelijk is aan p, dan moet de kans op de andere waarde wel gelijk zijn aan $1 - p$. Immers, kansen tellen op tot één. De informatie entropie is daarmee gemiddeld in een fractie p van alle instances gelijk aan $-\lg(p)$, en in een fractie $1 - p$ van alle instances gelijk aan $-\lg(1 - p)$. De verwachte *gemiddelde* entropie wordt dan gelijk aan

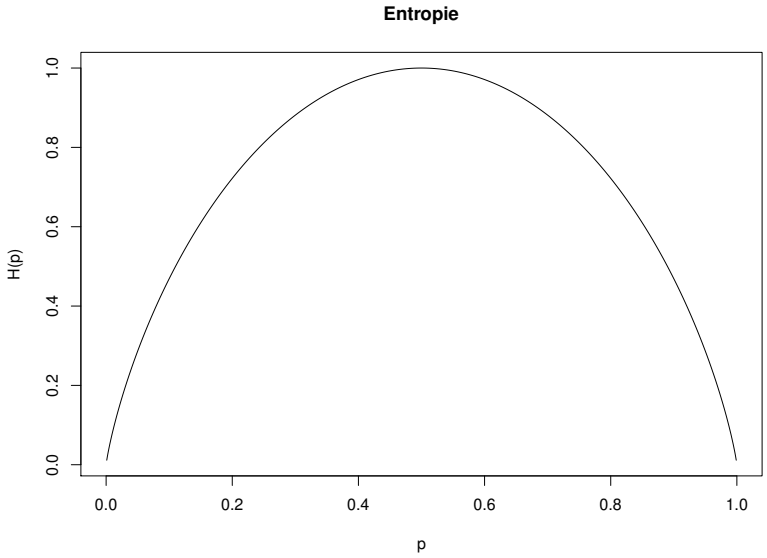
$$H = -p \cdot \lg(p) - (1 - p) \cdot \lg(1 - p)$$

Omdat dit de gemiddelde entropie is van één uitkomst wordt het resultaat van deze berekening uitgedrukt in "bits per instance".

Opgave 8. Stel, een oneerlijke munt heeft een kans van 55% om op "kop" te landen, en 45% kans op "munt". Is de gemiddelde entropie van de uitkomst van één worp in dit geval exact gelijk aan 1 bit, net als voor een eerlijke munt, of is deze groter of kleiner?

Opgave 9. Bereken met je rekenmachine de uitkomst voor H als $p = 0$. Wat gaat er mis? Voor welke andere waarde van p gaat het ook mis? Wat zou er eigenlijk uit moeten komen? Voor dit geval werkt de formule op je rekenmachine niet en zul je het resultaat moeten onthouden!

De onderstaande grafiek toont hoe de entropie afhangt van de kans p.



Een paar speciale gevallen verdienen aandacht:

- Voor de meest extreme gevallen $p = 0$ en $p = 1$ vinden we $H(p) = 0$. Dit is hopelijk niet meer verwonderlijk. In deze beide gevallen komt er feitelijk maar één waarde voor, want de kans op de andere waarde is nul. Zoals eerder gezegd weet je dan eigenlijk al tevoren wat de uitkomst is, en dus is de hoeveelheid informatie die met deze uitkomst gepaard gaat gelijk aan nul.
- Voor het geval $p = \frac{1}{2}$ dat daar exact tussenin ligt, bereikt de entropie een maximum met waarde $H(p) = 1$. Dit is niet vreemd omdat er precies één bit nodig is om de hoeveelheid informatie in een uitkomst te karakteriseren met twee even waarschijnlijk uitkomsten.

Opgave 10. Bereken of schat de informatie entropie van het antwoord op de vraag "is de kleur van je ogen blauw?"

Tot dusverre hadden we het over uitkomsten met twee mogelijkheden. In werkelijkheid kunnen vaak meerdere uitkomsten optreden; we stellen niet alleen ja/nee vragen. Voor nominale attributen met meer dan twee mogelijke waarden die elk met kansen $p_1, p_2, \ldots, p_k$ voorkomen, kan de bovenstaande formule veralgemeniseerd worden tot

$$H = -p_1 \cdot \lg(p_1) - p_2 \cdot \lg(p_2) - \ldots - p_k \cdot \lg(p_k)$$

Oftewel, met som-notatie,

$$H = \sum_{i=1}^{k} -p_i \cdot \lg(p_i)$$

De sommatie loopt hier over alle k mogelijke uitkomsten.

Ga voor jezelf na dat deze formule in het geval van $k = 2$ mogelijke uitkomsten precies de formule van de grafiek eerder hierboven oplevert.

Opgave 11. Een nucleotide-sequentie bevat vier verschillende nucleotiden: A, C, G en T. Als je mag aannemen dat elke nucleotide gemiddeld even vaak voorkomt, wat is dan de informatiedichtheid van zo'n sequentie? Schat het antwoord eerst en bereken het daarna exact.

De entropie van een uitkomst is het grootst als de kans op elke mogelijke uitkomst even groot is. Dat wil zeggen dat dan voor alle kansen geldt dat $p_i = \frac{1}{k}$. De bovenstaande formule levert in dat geval voor de entropie een maximale waarde gelijk aan $H = \lg(k)$. Als een uitkomst altijd dezelfde uitkomst heeft, bv. $p_1 = 1$ en alle andere $p_i = 0$, dan is de hoeveelheid informatie wederom minimaal en gelijk aan nul. Voor andere verdelingen ligt de hoeveelheid informatie hier tussenin. Kortom, voor een uitkomst met k mogelijke uitkomsten is $0 \leq H \leq \lg(k)$.

Bijvoorbeeld, voor een eerlijke dobbelsteen met zes verschillende uitkomsten die allemaal even waarschijnlijk zijn vinden we $H = \sum_{i=1}^{6} -\frac{1}{6} \cdot \lg(\frac{1}{6}) = 6 \cdot (-\frac{1}{6} \lg(\frac{1}{6})) = \lg(6) = 2.585$ bits/worp. Gevoelsmatig kan dit ook wel kloppen: met twee bits zijn $2^2 = 4$ verschillende uitkomsten te coderen, terwijl met drie bits $2^3 = 8$ verschillende uitkomsten mogelijk zijn; een dobbelsteen ligt hier ergens tussenin.

Opgave 12. Als het GC-ratio in een nucleotide-sequentie afwijkt van 0.50, wordt dan de entropie groter of kleiner dan in de vorige opgave, blijft deze hetzelfde, of is dit niet met zekerheid te zeggen?

Opgave 13. Bekijk de "contact-lenses" arff-datafile van Weka. Deze bevat 24 instances met drie verschillende klassen: "soft", "hard", en "none". Bepaal op grond van de relatieve frequenties van deze drie klassen de entropie van het klasselabel.

Opgave 14. In de mens bevat een eiwit typisch 375 aminozuren. Deze hebben een gemiddelde relatieve frequentie volgens onderstaande tabel. Bepaal de *totale* hoeveelheid informatie die nodig is om een typische eiwitsequentie te beschrijven.

ALA	7.4%	GLU	5.8%	LEU	7.6%	SER	8.1%
ARG	4.2%	GLN	3.7%	LYS	7.2%	THR	6.2%
ASN	4.4%	GLY	7.4%	MET	1.8%	TRP	1.3%
ASP	5.9%	HIS	2.9%	PHE	4.0%	TYR	3.3%
CYS	3.3%	ILE	3.8%	PRO	5.0%	VAL	6.7%

2 Information gain & Gain ratio

Classificatie-algoritmen die op boomstructuren (*trees*) zijn gebaseerd, passen herhaaldelijk een attribuut toe om tot een classificatie te komen. Ze zijn daarmee te beschouwen als recursieve varianten van *OneR*. Het doel van een tree is om uiteindelijk homogene leafnodes te verkrijgen waarbinnen alle instances tot dezelfde klasse behoren. De entropie is dan afgenomen tot nul. Dit doel kan overigens niet altijd worden bereikt, bijvoorbeeld als er te weinig attributen zijn om tot een volledige scheiding te komen.

Deze tree-algoritmen benutten veelal entropie om te bepalen welk attribuut telkens gebruikt moet worden om een classificatie verder te verfijnen. Daartoe wordt enerzijds

bepaald hoeveel entropie de instances hebben wanneer ze een node binnen een tree ingaan, en anderzijds hoeveel entropie de instances gemiddeld hebben wanneer ze de node weer verlaten. Het verschil hiertussen wordt de *information gain* genoemd.

Neem bijvoorbeeld Weka's weather.nominal dataset. Bij aanvang zijn er 14 instances, waarvan 9 met het klasselabel "yes" en 5 met het klasselabel "no". De entropie van deze verdeling (in bits/instance) is

$$H_{\text{in}} = \sum_{i=1}^{k} -p_i \cdot \lg(p_i) = -\frac{9}{14}\lg(\frac{9}{14}) - \frac{5}{14}\lg(\frac{5}{14}) = 0.940$$

Stel, we zouden vervolgens opsplitsen op grond van het attribuut Outlook. Nu zijn er drie mogelijkheden:

- Volgen we de tak "sunny", dan blijven er vijf instances over, waarvan twee met het label "yes" en drie met het label "no". Dit levert een entropie van $H_{\text{sunny}} = -\frac{2}{5}\lg(\frac{2}{5}) - \frac{3}{5}\lg(\frac{3}{5}) = 0.971$ bits/instance op.
- Volgen we de tak "overcast", dan blijven er vier instances over, alle met het label "yes". Dit levert een entropie van $H_{\text{overcast}} = -\frac{4}{4}\lg(\frac{4}{4}) - \frac{0}{4}\lg(\frac{0}{4}) = 0.000$ bits/instance op. (Op je rekenmachine gaat het mis, maar onthoud dat $0 \cdot \lg(0) = 0$.)
- Volgen we de tak "rainy", dan blijven er vijf instances over, waarvan drie met het label "yes" en twee met het label "no". Dit levert een entropie van $H_{\text{rainy}} = -\frac{3}{5}\lg(\frac{3}{5}) - \frac{2}{5}\lg(\frac{2}{5}) = 0.971$ bits/instance op.

Om te bepalen hoeveel entropie na de opsplitsing resteert, gemiddeld over alle instances, dienen we deze drie uitkomsten gewogen te middelen: we hebben $\frac{5}{14}$ kans op een "sunny" instance met entropie 0.971, $\frac{4}{14}$ kans op een "overcast" instance met entropie 0.000, en $\frac{5}{14}$ kans op een "rainy" instance met entropie 0.971. Dit geeft in totaal

$$H_{\text{uit}} = \frac{5}{14} \cdot 0.971 + \frac{4}{14} \cdot 0.000 + \frac{5}{14} \cdot 0.971 = 0.694$$

wat dus neerkomt op een information gain van

$$\Delta H = H_{\text{in}} - H_{\text{uit}} = 0.940 - 0.694 = 0.247$$

Een soortgelijke berekening kan worden uitgevoerd voor de attributen Temperature, Humidity en Windy. Het uiteindelijke resultaat luidt als volgt (steeds in bits/instance):

Attribuut	H_{in}	H_{uit}	ΔH
Outlook	0.940	0.693	0.247
Temperature	0.940	0.911	0.029
Humidity	0.940	0.788	0.152
Windy	0.940	0.892	0.048

Opgave 15. Voer zelf de berekening uit voor de attributen Temperature, Humidity en Windy, en controleer dat je op dezelfde antwoorden uitkomst als in de tabel hierboven.

Het attribuut waarvan we het meeste leren is het attribuut dat tot de grootste information gain leidt. Als je één attribuut moet kiezen kun je het beste op splitsen dat attribuut. Voor deze dataset dient dus te worden gesplitst op Outlook. Dit levert immers de grootste information gain ΔH en laat de entropie H_{uit} het verste dalen richting nul. De geproduceerde takkan zijn daarmee zo homogeen mogelijk.

Voor de uitgaande tak met waarde "overcast" hoeft daarna niets meer te gebeuren: deze leaf-node is al homogeen omdat alle vier de instances reeds hetzelfde klasselabel hebben. Voor de takken behorende bij een "sunny" en "rainy" outlook is dit (nog) niet het geval en dient hetzelfde recept nogmaals recursief te worden herhaald.

Opgave 16. Ga na welke attributen in de "sunny" en "rainy" takken daarna de beste splitsing opleveren. Lukt het om de leaf-nodes homogeen te maken? En komen je eigen antwoorden overeen met het model dat de *Id3-tree* classifier in Weka produceert?

Er treedt een complicatie op wanneer er attributen zijn met unieke waarden voor alle instances. Stel bijvoorbeeld dat in deze dataset ook de datum van de veertien instances was gegeven. Dan zou splitsen op dit datum-attribuut veertien individuele takken geven waarbij telkens één instance zou horen. Dan zou in één keer elke leaf-node homogeen zijn, wat vanzelfsprekend de grootst mogelijke information gain zou opleveren. Tegelijkertijd zou dit niet een "zinvolle" manier van opsplitsen zijn omdat deze niet generaliseert naar nieuwe data.

Zelfs als er niet een dergelijke extreme situatie optreedt heeft het zin om rekening te houden met het aantal mogelijke vertakkingen dat een attribuut oplevert. Een geschikte manier om hiervoor te corrigeren is door per attribuut tevens de entropie van de splitsing te bepalen. Bijvoorbeeld, voor het attribuut Outlook dat we hierboven hebben uitgewerkt vond er een opsplitsing plaats in drie takken ("sunny", "overcast" en "rainy") met daarin respectievelijk 5, 4 en 5 instances. De entropie van deze opsplitsing is

$$H_{\text{split}} = -\frac{5}{14}\lg(\frac{5}{14}) - \frac{4}{14}\lg(\frac{4}{14}) - \frac{5}{14}\lg(\frac{5}{14}) = 1.577$$

Naarmate de opsplitsing fijnmaziger is, is de uitkomst van deze berekening hoger. We kunnen de information gain ΔH corrigeren voor het effect van deze splitsing door te delen door deze waarde H_{split}. We verkrijgen dan het *gain ratio*. Voor het attribuut Outlook geeft dit

$$h = \frac{\Delta H}{H_{\text{split}}} = \frac{0.247}{1.577} = 0.156$$

Voeren we deze berekening uit voor alle attributen in de root-node, dan vinden we de volgende uitkomsten:

Attribuut	H_{in}	H_{uit}	ΔH	H_{split}	h
Outlook	0.940	0.694	0.247	1.577	0.156
Temperature	0.940	0.911	0.029	1.557	0.019
Humidity	0.940	0.788	0.152	1.000	0.152
Windy	0.940	0.892	0.048	0.985	0.049

Opgave 17. Bepaal zelf H_{split} en het gain ratio h voor de attributen Temperature, Humidity en Windy, en controleer dat je op dezelfde antwoorden uitkomst als in de tabel hierboven.

Weliswaar wint het attribuut Outlook het nu nog steeds, maar het attribuut Humidity komt opeens wel heel erg dicht daarbij in de buurt op de tweede plaats. Humidity is in deze ranglijst gestegen omdat het in staat is een redelijk goede opsplitsing te verkrijgen op grond van slechts twee mogelijke waarden ("normal" en "high"), terwijl Outlook er daar drie voor nodig had.

Opgave 18. Stel dat er hierboven een datum-attribuut aanwezig was geweest met voor elke instance een andere waarde. Bereken voor dit attribuut eveneens de grootheden H_{in}, H_{uit}, ΔH, H_{split} en h, en kijk hoe goed dit nieuwe attribuut scoort vergeleken met de andere vier attributen.

Waar het *Id3-tree* algoritme gebruik maakt van information gain om voor elke node in de tree een attribuut te kiezen, gebruikt het *J48-tree* algoritme het gain ratio.

Opgave 19. Bepaal voor de "contact-lenses" arff-datafile handmatig de root-node voor zowel de *Id3-* als de *J48-tree*.