Deel III

Probabilistische Modellen

Les 11 Onzekerheid, entropie en informatie

Als we erover nadenken hoe we conclusies trekken, komen we er snel achter dat dit meestal met het verkrijgen en verwerken van informatie te maken heeft. Vaak stellen we hiervoor vragen of maken een meting, om de onzekerheid die we over benodigde gegevens hebben te overkomen of tenminste te verkleinen.

Als we nu op het gebied van de kunstmatige intelligentie een systeem willen bouwen, dat op grond van zekere informatie beslissingen neemt, moeten we voor de begrippen *informatie* of *onzekerheid* definities vinden, die het mogelijk maken om ook kwantitatieve uitspraken hierover te kunnen doen. Een cruciaal begrip in dit kader is de *entropie* van een kansverdeling, die in principe aangeeft hoeveel bits we minstens nodig hebben, om de uitkomsten van een kansexperiment te beschrijven.

11.1 Onzekerheid

Als we een experiment of gebeurtenis door een kansverdeling beschrijven, drukken we hiermee uit dat we niet zeker over de uitkomst zijn. Maar we hebben ook een intuïtieve idee dat de onzekerheid soms groter is dan in andere gevallen. Bijvoorbeeld zijn we onzekerder over de uitkomst bij het werpen van een dobbelsteen dan bij het werpen van een munt, omdat er in het ene geval 6 mogelijke uitkomsten zijn, maar in het andere geval slechts 2. Ook bij een sportwedstrijd hangt onze onzekerheid ervan af hoe we de kansen voor de uitkomst inschatten: Als alleen maar de KI-studenten onderling een zwemwedstrijd uitvechten is de onzekerheid waarschijnlijk groter dan als Pieter van den Hoogenband ook meedoet.

Voorbeeld: Stel bij een paardenrace doen 8 paarden mee die niet even sterk zijn, maar waarvoor de kansen om te winnen gegeven zijn door

$$p_1 = \frac{1}{2}, \quad p_2 = \frac{1}{4}, \quad p_3 = \frac{1}{8}, \quad p_4 = \frac{1}{16}, \quad p_5 = p_6 = p_7 = p_8 = \frac{1}{64}.$$

Als we gewoon het nummer van het winnende paard door willen geven, hebben we hiervoor 3 bits nodig (want $2^3 = 8$). Maar omdat de kansen niet uniform verdeeld zijn, kunnen dit aantal reduceren, door de paarden met een hogere kans op een kortere manier te coderen. Hierbij moeten we er wel op letten, dat de beginstukken van de langere coderingen zelfs geen coderingen zijn. Een mogelijke codering voor de nummers 1 t/m 8 van de paarden in het voorbeeld is (aangegeven met strings van bits):

1: `0`, 2: `10`, 3: `110`, 4: `1110`, 5: `111100`, 6: `111101`, 7: `111110`, 8: `111111`.

Als we voor deze codering het gemiddeld benodigde aantal bits berekenen (dus de verwachtingswaarde van het aantal bits), krijgen we $\frac{1}{2} \cdot 1 + \frac{1}{4} \cdot 2 + \frac{1}{8} \cdot 3 + \frac{1}{16} \cdot 4 + 4 \cdot \frac{1}{64} \cdot 6 = 2$. We hebben het aantal benodigde bits dus van 3 op 2 kunnen reduceren, door de uitkomsten die we vaker verwachten korter te coderen.

De onzekerheid bij een kansexperiment is natuurlijk bepaald door de kansen die we aan de mogelijke uitkomsten toewijzen. We kunnen ons dus afvragen

hoe we voor een discrete kansverdeling $P = (p_1, \ldots, p_n)$ een waarde voor de onzekerheid kunnen berekenen. Het idee dat we hiervoor hebben, is een functie

$$H(P) = H(p_1, \ldots, p_n)$$

te vinden, die de onzekerheid weergeeft. Omdat we intuïtief wel een idee van de onzekerheid bij een kansverdeling hebben, moet zo'n functie zekere eigenschappen hebben. In het jaar 1948 is hiervoor door C.E. Shannon (dezelfde Shannon als bij het sampling theorema) in het kader van de communicatietheorie een voorstel gedaan aan welke eisen zo'n functie $H(P)$ zou moeten voldoen. De link tussen communicatietheorie en kansrekening bestaat erin, dat communicatie als transmissie (van bit-strings, dus van ketens van 0en en 1en) via kanalen gemodelleerd wordt, waarbij er toevallig fouten kunnen optreden. De vraag is dan, hoe veel onzekerheid in het ontvangen signaal ligt.

Eisen aan een functie voor de onzekerheid van een kansverdeling

De eisen die Shannon heeft gesteld zijn als volgt:

(1) $H(P)$ is een continue functie in de argumenten $p_1, \ldots, p_n$, want als we de kansen maar heel weinig veranderen, verandert ook de onzekerheid nauwelijks.

(2) De onzekerheid hangt alleen maar van de kansen p_i, maar niet van hun volgorde af, dus geldt $H(p_1, \ldots, p_n) = H(p_{\pi(1)}, \ldots, p_{\pi(n)})$ voor elke permutatie π van de indices.

(3) $H(P) \geq 0$ en $H(P) = 0$ alleen maar als één van de $p_i = 1$ is (en de anderen dus 0). Dit betekent dat we altijd onzeker zijn, behalve als een uitkomst kans 1 heeft en dus zeker gaat gebeuren.

(4) $H(p_1, \ldots, p_n) = H(p_1, \ldots, p_n, 0)$, dus de onzekerheid verandert niet, als we de kansverdeling uitbreiden tot meer mogelijke gebeurtenissen, maar de nieuwe opties kans 0 hebben en dus nooit kunnen gebeuren.

(5) $H(\frac{1}{n}, \ldots, \frac{1}{n}) \leq H(\frac{1}{n+1}, \ldots, \frac{1}{n+1})$, d.w.z. de onzekerheid bij een uniforme verdeling met $n+1$ mogelijke uitkomsten is groter dan bij n mogelijke uitkomsten.

(6) $H(\frac{1}{mn}, \ldots, \frac{1}{mn}) = H(\frac{1}{m}, \ldots, \frac{1}{m}) + H(\frac{1}{n}, \ldots, \frac{1}{n})$. Als we twee onafhankelijke experimenten met uniforme verdelingen tot een gezamenlijk experiment combineren, willen we dat de onzekerheid van het gecombineerde experiment juist de som van de onzekerheden bij de enkele experimenten is.

(7) We splitsen de verzameling $\Omega = \{1, \ldots, n\}$ op in de twee deelverzamelingen $\Omega_1 = \{1, \ldots, r\}$ en $\Omega_2 = \{r+1, \ldots, n\}$. De totale kans voor de uitkomsten in Ω_1 is $q_1 = p_1 + \ldots + p_r$ en de kans voor Ω_2 is $q_2 = p_{r+1} + \ldots + p_n$. De onzekerheid of een uitkomst in Ω_1 of Ω_2 ligt, is $H(q_1, q_2)$, de onzekerheid over een uitkomst in Ω_1 is $H(\frac{p_1}{q_1}, \ldots, \frac{p_r}{q_1})$, omdat $(\frac{p_1}{q_1}, \ldots, \frac{p_r}{q_1})$ juist de

kansverdeling op Ω_1 is. Net zo is $H(\frac{p_{r+1}}{q_2}, \ldots, \frac{p_n}{q_2})$ de onzekerheid over een uitkomst in Ω_2. De totale onzekerheid over de uitkomst van P is samengesteld uit de onzekerheden in welke deelverzameling een uitkomst ligt en de onzekerheden van de twee deelverzamelingen, die met hun kansen q_1 en q_2 gewogen zijn, dus moet gelden:

$$H(p_1, \ldots, p_n) = H(q_1, q_2) + q_1 H(\frac{p_1}{q_1}, \ldots, \frac{p_r}{q_1}) + q_2 H(\frac{p_{r+1}}{q_2}, \ldots, \frac{p_n}{q_2}).$$

De meeste van deze punten zijn volstrekt intuïtief, alleen de punten (6) en (7) stellen inhoudelijke eisen, namelijk hoe de onzekerheden van verschillende gebeurtenissen gecombineerde moeten worden.

Het interessante (en misschien verrassende) is nu, dat deze eisen zo sterk zijn dat er in principe alleen maar een enkele functie $H(P)$ bestaat die aan de eisen voldoet, namelijk de functie:

$$H(P) = H(p_1, \ldots, p_n) = -\lambda \sum_{i=1}^{n} p_i \log(p_i)$$

met $\lambda > 0$, waarbij de som alleen maar over de p_i met $p_i \neq 0$ loopt.

We zullen dit hier niet bewijzen, maar wel toelichten dat de functie $H(P)$ aan de eisen (1)-(7) voldoet. Hierbij zijn de punten (1)-(4) rechtstreeks duidelijk, de andere punten gaan we even na. Omdat de constante λ geen enkel verschil in de argumenten maakt, werken we voor het gemak met $\lambda = 1$.

(5) Voor een uniforme verdeling U_n op n punten geldt

$$H(U_n) = -\sum_{i=1}^{n} \frac{1}{n} \log(\frac{1}{n}) = \sum_{i=1}^{n} \frac{1}{n} \log(n) = \log(n)$$

en omdat $\log(x)$ een strikt stijgende functie is, is $H(U_n) = \log(n) < \log(n+1) = H(U_{n+1})$.

(6) Dit volgt ook uit het feit dat $H(U_n) = \log(n)$, omdat $\log(mn) = \log(m) + \log(n)$.

(7) Uit $q_1 = \sum_{i=1}^{r} p_i$ en $q_2 = \sum_{i=r+1}^{n} p_i$ volgt

$$\begin{aligned}
&H(q_1, q_2) + q_1 H(\frac{p_1}{q_1}, \ldots, \frac{p_r}{q_1}) + q_2 H(\frac{p_{r+1}}{q_2}, \ldots, \frac{p_n}{q_2}) \\
&= -q_1 \log(q_1) - q_2 \log(q_2) - q_1 \sum_{i=1}^{r} \frac{p_i}{q_1} \log(\frac{p_i}{q_1}) - q_2 \sum_{i=r+1}^{n} \frac{p_i}{q_2} \log(\frac{p_i}{q_2}) \\
&= -\sum_{i=1}^{r} p_i \log(q_1) - \sum_{i=r+1}^{n} p_i \log(q_2) - \sum_{i=1}^{r} p_i(\log(p_i) - \log(q_1)) \\
&\quad - \sum_{i=r+1}^{n} p_i(\log(p_i) - \log(q_2)) \\
&= -\sum_{i=1}^{n} p_i \log(p_i) = H(p_1, \ldots, p_n).
\end{aligned}$$

Als we ons afvragen, bij welke kansverdeling met n mogelijke uitkomsten we de grootste onzekerheid hebben, ligt het voor de hand dat dit bij een uniforme verdeling het geval is, want in dit geval hebben we geen reden om een voorkeur aan een of andere uitkomst te geven. Als $H(P)$ een maat voor de onzekerheid is, zouden we dus verwachten dat de waarde van de functie $H(P)$ voor een uniforme verdeling maximaal is en dit laat zich inderdaad bewijzen.

Stelling: Van alle kansverdelingen P op n mogelijke uitkomsten geeft de uniforme verdeling met $p_i = \frac{1}{n}$ de maximale waarde van de functie $H(P)$.

Omdat het bewijs van deze stelling niet moeilijk is en belangrijke inzichten geeft, gaan we het even na:

In het punt $x = 1$ is $\log(x) = 0$ en $\log'(x) = 1$, dus is de lijn met vergelijking $y = x - 1$ de raaklijn aan de grafiek van de logaritme in het punt $x = 1$. Omdat $\log''(x) = -\frac{1}{x^2} < 0$, blijft de logaritme steeds onder deze raaklijn, daarom geldt

$$\log(x) \leq x - 1 \text{ met gelijkheid alleen maar voor } x = 1.$$

Voor twee kansverdelingen $P = (p_1, \ldots, p_n)$ en $Q = (q_1, \ldots, q_n)$ volgt hieruit dat

$$\sum_{i=1}^{n} p_i \log(\frac{q_i}{p_i}) \leq \sum_{i=1}^{n} p_i(\frac{q_i}{p_i} - 1) = \sum_{i=1}^{n} q_i - \sum_{i=1}^{n} p_i = 1 - 1 = 0.$$

Wegens $\log(\frac{q_i}{p_i}) = \log(q_i) - \log(p_i)$ volgt hieruit dat

$$-\sum_{i=1}^{n} p_i \log(p_i) \leq -\sum_{i=1}^{n} p_i \log(q_i).$$

Als we nu voor Q speciaal de uniforme verdeling U_n met $q_i = \frac{1}{n}$ kiezen, volgt hieruit aan de ene kant dat

$$H(P) \leq -\sum_{i=1}^{n} p_i \log(\frac{1}{n}) = \sum_{i=1}^{n} p_i \log(n) = \log(n).$$

Maar aan de andere kant is $H(U_n) = -\sum_{i=1}^{n} \frac{1}{n} \log(\frac{1}{n}) = \log(n)$, dus is de waarde voor de uniforme verdeling inderdaad maximaal.

We hebben inmiddels twee belangrijke inzichten gewonnen, die we nog eens expliciet willen aangeven:

(I) Voor een uniforme verdeling U_n op n punten is $H(U_n) = \log(n)$.

(II) Voor twee kansverdelingen P en Q is $-\sum p_i \log(q_i) \geq H(P)$ en er geldt

$$D(P, Q) := \sum_{i=1}^{n} p_i \log(\frac{p_i}{q_i}) \geq 0$$

want we hadden gezien dat $\sum_{i=1}^{n} p_i \log(p_i) \geq \sum_{i=1}^{n} p_i \log(q_i)$ en hieruit volgt $\sum_{i=1}^{n} p_i(\log(p_i) - \log(q_i)) = \sum_{i=1}^{n} p_i \log(\frac{p_i}{q_i}) \geq 0$.

Omdat de ideeën voor het formaliseren van onzekerheid uit de communicatietheorie komen waar men het over bit-strings heeft, is het gebruikelijk de functie $H(P)$ niet met behulp van de natuurlijke logaritme (met basis e) maar met de logaritme met basis 2 te formuleren. Wegens ${}^2\log(x) = \frac{\log(x)}{\log(2)}$ geeft dit alleen maar een verschil van de constante factor $\log(2)$.

Definitie: De functie

$$H(P) = H(p_1, \ldots, p_n) := -\sum_{i=1}^{n} p_i \ {}^2\log(p_i)$$

heet de *entropie* van de kansverdeling P.

> Het begrip *entropie* speelt ook in de natuurkunde, vooral in de thermodynamica, een belangrijke rol. Hier geeft de entropie een maat voor de wanorde in een systeem. De tweede hoofdstelling van de thermodynamica zegt (in het grof) dat in een gesloten systeem de entropie nooit afneemt, d.w.z. dat zonder invloed van buiten de wanorde in een systeem steeds toeneemt. (Dit is natuurlijk ook een alledaagse ervaring.)

We hebben tot nu toe de entropie alleen maar voor een kansverdeling gedefinieerd. Vaak spreekt men immers ook van de entropie van een stochast X. Hiermee is de entropie van de kansverdeling van de mogelijke uitkomsten van X bedoelt. Stel een stochast X heeft de mogelijke uitkomsten $x_1, \ldots, x_n$, dan geeft $p_i := p(X = x_i)$ de kans op de i-de mogelijke uitkomst en de kansverdeling $P = (p_1, \ldots, p_n)$ beschrijft de kansen van de mogelijke uitkomsten van X. We definiëren dus de de entropie van een stochast X met mogelijke uitkomsten $x_1, \ldots, x_n$ door

$$H(X) := -\sum_{i=1}^{n} p(X = x_i) \ {}^2\log(p(X = x_i)).$$

Voorbeeld: Zij X de stochast van een Bernoulli experiment met kans p op succes (uitkomst 1) en kans $1 - p$ op mislukken (uitkomst 0). Er geldt

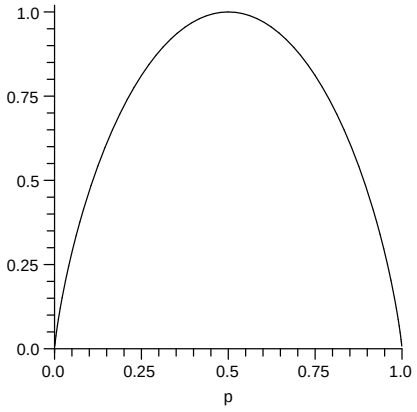
$$\begin{aligned} H(X) &= -p(X = 1) \ {}^2\log(p(X = 1)) - p(X = 0) \ {}^2\log(p(X = 0)) \\ &= -p \ {}^2\log(p) - (1 - p) \ {}^2\log(1 - p). \end{aligned}$$

In Figuur III.1 is duidelijk te zien dat de entropie maximaal wordt voor $p = 0.5$, dus voor een uniforme verdeling en dat in dit geval de entropie juist 1 *bit* is.

Relatieve entropie en de Kullback-Leibler afstand

We hebben gezien dat voor twee kansverdelingen $P = (p_1, \ldots, p_n)$ en $Q = (q_1, \ldots, q_n)$ geldt dat

$$D(P, Q) := \sum p_i({}^2\log(p_i) - {}^2\log(q_i)) = \sum p_i \ {}^2\log(\frac{p_i}{q_i}) \geq 0$$



Figuur III.1: Entropie van een Bernoulli experiment afhankelijk van de kans p op succes.

met gelijkheid alleen maar als $p_i = q_i$ voor alle i. Men noemt $D(P,Q)$ de *relatieve entropie* of *Kullback-Leibler afstand* tussen P en Q.

De relatieve entropie $D(P,Q)$ geeft aan, hoe veel bits we gemiddeld extra nodig hebben, omdat we de codering van de gegevens op grond van de (verkeerde) kansverdeling Q in plaats van P hebben gekozen. Er geldt namelijk

$$H(P) + D(P,Q) = -\sum_{i=1}^{n} p_i \ {}^2\log(p_i) + \sum_{i=1}^{n} p_i \ {}^2\log(\frac{p_i}{q_i}) = -\sum_{i=1}^{n} p_i \ {}^2\log(q_i)$$

en dit is juist de verwachtingswaarde van het aantal benodigde bits op grond van de kansverdeling Q.

Merk op: De naam Kullback-Leibler *afstand* voor de relatieve entropie is een beetje misleidend, omdat we het niet met een afstand zo als de gewone Euclidische afstand in het vlak of in de ruimte te maken hebben.

Een echte afstandsfunctie moet namelijk de volgende drie eigenschappen hebben:

(i) $d(P,Q) \geq 0$ en $d(P,Q) = 0$ alleen maar als $P = Q$,

(ii) $d(P,Q) = d(Q,P)$ (symmetrie),

(iii) $d(P,Q) + d(Q,R) \geq d(P,R)$ (driehoeksongelijkheid).

De relatieve entropie heeft alleen maar de eerste van deze drie eigenschappen. Maar met een eenvoudig trucje kunnen we van de relatieve entropie wel een symmetrische functie maken, namelijk door

$$d_{KL}(P,Q) := \frac{1}{2}(D(P,Q) + D(Q,P)) = \frac{1}{2}\sum p_i \ {}^2\log(\frac{p_i}{q_i}) + q_i \ {}^2\log(\frac{q_i}{p_i}).$$

Ook dit heet meestal de Kullback-Leibler afstand van P en Q, soms iets duidelijker de *symmetrische Kullback-Leibler afstand*.

Ook al voldoet de Kullback-Leibler niet aan de driehoeksongelijkheid, zijn $D(P,Q)$ of $d_{KL}(P,Q)$ toch vaak handig om te kwantificeren hoe sterk verschillende kansverdelingen Q op een vaste (doel-)kansverdeling P lijken.

11.2 Entropie van continue kansverdelingen

We hebben ons tot nog toe tot discrete kansverdelingen beperkt. De overgang tot continue kansverdeling is echter geen probleem: In plaats van de kansen p_i krijgen we een dichtheidsfunctie $f(x)$ voor de kansverdeling en de som over de mogelijke uitkomsten wordt de integraal over de continue variabel x. Voor de entropie van een stochast X met dichtheidsfunctie $f(x)$ krijgt men zo:

$$H(X) := -\int_{-\infty}^{\infty} f(x)\ {}^2\log(f(x))\ dx.$$

> Om duidelijk te maken dat het om de entropie van een continue variabel gaat, spreekt men vaak ook van *differentiële entropie*. Het idee achter deze naam is, de variabel x te discretiseren door de waarden in het interval $[x_i - \frac{\Delta x}{2}, x_i + \frac{\Delta x}{2}]$ aan de discrete waarde x_i toe te wijzen en de kans over dit interval als kans $p_i := \int_{x_i-\frac{\Delta x}{2}}^{x_i+\frac{\Delta x}{2}} f(x)\ dx$ te definiëren. Met de overgang $\Delta x \to 0$ komt men dan naar de continue versie van de entropie.

Ook de *relatieve entropie* of *Kullback-Leibler afstand* van twee stochasten X en Y met dichtheidsfuncties $f(x)$ en $g(x)$ wordt analoog met het discrete geval gedefinieerd, namelijk door

$$D(X,Y) := \int_{-\infty}^{\infty} f(x)\ {}^2\log(\frac{f(x)}{g(x)})\ dx.$$

Met hetzelfde argument als bij de discrete kansverdelingen geldt weer

$$D(X,Y) \geq 0 \quad \text{en} \quad \int_{-\infty}^{\infty} f(x)\ {}^2\log(g(x))\ dx = H(X) + D(X,Y).$$

Men ziet makkelijk in dat de entropie van een stochast X onafhankelijk van de verwachtingswaarde $\mu := E[X]$ is, want met de substitutie $x' = x + a$ volgt dat de verschoven stochast $X + a$ dezelfde entropie als X heeft. Aan de andere kant heeft de variantie $Var(X) = E[X^2 - E[X]^2]$ zeker een invloed op de entropie, want hoe sterker de resultaten vaan X verspreid zijn, hoe onzekerder zijn we over de uitkomsten van X.

Bij discrete kansverdelingen hadden we gezien, dat onder de verdelingen met n mogelijke uitkomsten de uniforme verdeling de hoogste entropie heeft. De equivalente vraag voor continue kansverdelingen is, welke verdeling met gegeven variantie σ^2 de grootste entropie heeft.

Het lijkt misschien enigszins verrassend dat we ook deze vraag kunnen beantwoorden, want we moeten een uitspraak over alle mogelijke dichtheidsfuncties maken. Maar er laat zich aantonen dat bij gegeven variantie de normale

verdeling de maximale entropie heeft, dus dat we bij de normale verdeling de grootste onzekerheid over de mogelijke uitkomsten hebben.

Stelling: Onder alle continue kansverdelingen met variantie σ^2 heeft de normale verdeling

$$f(x) = \frac{1}{\sqrt{2\pi}\,\sigma}\; e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

de maximale entropie.

> Het idee onder zekere randvoorwaarden de kansverdeling met maximale entropie te bepalen, geeft aanleiding tot een alternatieve manier om parameters van een probabilistisch model te schatten. In Wiskunde 1 hadden we hiervoor al de *maximum likelihood* methode leren kennen, waarbij de parameters zo bepaald worden dat de kans op de waargenomen resultaten maximaal is. Bij de toegang middels maximale entropie worden de parameters zo gekozen, dat de entropie maximaal wordt, dus het wordt het meest algemene model verondersteld dat de waarnemingen verklaart.
>
> Vaak is een algemeen model ook eenvoudiger dan een speciaal model en heeft het voordeel enigszins robuust tegen uitschieters in het training materiaal te zijn. Het principe om onder gegeven randvoorwaarden het eenvoudigste model te kiezen staat ook bekend onder de naam *Ockham's razor* (na de filosoof William van Ockham (1285-1349): 'The simplest explanation is the best.').

Voor het bewijs van de stelling dat de normale verdeling de maximale entropie heeft, is een techniek namens *variatierekening* nodig. Hierbij gaat het om het vinden van extrema van functies, die niet van een of meerdere variabelen afhangen maar van een continue hoeveelheid variabelen, anders gezegd om maxima en minima van functies, die zelfs ook weer van functies afhangen. We zullen in deze cursus geen variatierekening behandelen, maar schetsen wel even het idee.

In ons geval willen we een maximum van de functie

$$H(f) = -\int_{-\infty}^{\infty} f(x)\ {}^2\log(f(x))\ dx$$

vinden, die van de dichtheidsfunctie $f = f(x)$ afhangt. Hierbij moet $f(x)$ aan zekere randvoorwaarden voldoen, namelijk dat het een dichtheidsfunctie is, dat de variantie σ^2 is, en we mogen nog veronderstellen dat de verwachtingswaarde $\mu = 0$ is. We moeten dus een maximum van $H(f)$ vinden onder de randvoorwaarden:

(i) $f(x) \geq 0$;

(ii) $\int_{-\infty}^{\infty} f(x)\ dx = 1$;

(iii) $\int_{-\infty}^{\infty} x f(x)\ dx = 0$;

(iv) $\int_{-\infty}^{\infty} x^2 f(x)\, dx = \sigma^2$.

Dit is natuurlijk typisch een situatie voor Lagrange multiplicatoren, we definiëren daarom de Lagrange functie

$$L(f) = -\int_{-\infty}^{\infty} f(x)\log(f(x))\, dx + \lambda_0(\int_{-\infty}^{\infty} f(x)\, dx - 1) + \lambda_1(\int_{-\infty}^{\infty} xf(x)\, dx) + \lambda_2(\int_{-\infty}^{\infty} x^2 f(x)\, dx - \sigma^2).$$

Hierbij vergeten we even de randvoorwaarde $f(x) \geq 0$, die zal uiteindelijk van zelfs goed komen. We werken met de natuurlijke logaritme $\log(x)$ in plaats van de logaritme met basis 2, omdat dit voor het bepalen van de afgeleiden handiger is.

Om de kritieke punten van de Lagrange functie $L(f)$ te vinden, moeten we nu de partiële afgeleiden naar de variabelen bepalen, dus naar de functiewaarden $f(x)$ van de dichtheidsfunctie. Merk op dat x in dit geval een constante en geen variabel is, de variabelen zijn juist de functiewaarden op gegeven punten x. We moeten nu $L(f)$ voor een vaste x naar $f(x)$ afleiden en dit gelijk aan 0 zetten. Omdat we hierbij alleen maar naar een enkele waarde van x kijken, mogen we de integralen in $L(f)$ meteen weglaten. We krijgen

$$\begin{aligned}\frac{\partial L}{\partial f(x)} &= -\log(f(x)) - f(x)\cdot\frac{1}{f(x)} + \lambda_0\cdot 1 + \lambda_1\cdot x + \lambda_2\cdot x^2 \\ &= -\log(f(x)) - 1 + \lambda_0 + \lambda_1 x + \lambda_2 x^2.\end{aligned}$$

Uit $\frac{\partial L}{\partial f(x)} = 0$ volgt nu $\log(f(x)) = -1 + \lambda_0 + \lambda_1 x + \lambda_2 x^2$ en dus

$$f(x) = e^{-1+\lambda_0+\lambda_1 x+\lambda_2 x^2}.$$

Maar dit betekent dat $f(x)$ juist een normale verdeling is en volgens de randvoorwaarden moeten de constanten λ_0, λ_1, λ_2 zo gekozen worden dat de verwachtingswaarde 0 en de variantie σ^2 wordt, en dit is juist voor

$$f(x) = \frac{1}{\sqrt{2\pi}\,\sigma}\, e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

het geval.

We hebben tot nu toe alleen maar aangetoond dat de normale verdeling een kritieke waarde voor de Lagrange functie is. Maar als we nu veronderstellen dat $g(x)$ de dichtheidsfunctie van een stochast Y met verwachtingswaarde 0 en variantie σ^2 is, kunnen we aantonen dat $H(Y) \leq H(X)$ is, dus dat de entropie

voor de normale verdeling inderdaad een maximum aanneemt:

$$\begin{aligned}
H(Y) &= -\int_{-\infty}^{\infty} g(x)\log(g(x))\,dx = -\int_{-\infty}^{\infty} g(x)\log(\frac{g(x)}{f(x)}\cdot f(x))\,dx \\
&= -\int_{-\infty}^{\infty} g(x)\log(\frac{g(x)}{f(x)})\,dx - \int_{-\infty}^{\infty} g(x)\log(f(x))\,dx \\
&= -D(Y,X) - \int_{-\infty}^{\infty} g(x)\log(f(x))\,dx \\
&\leq_{(*)} -\int_{-\infty}^{\infty} g(x)(-1+\lambda_0+\lambda_1 x+\lambda_2 x^2)\,dx \\
&=_{(**)} -\int_{-\infty}^{\infty} f(x)(-1+\lambda_0+\lambda_1 x+\lambda_2 x^2)\,dx \\
&= -\int_{-\infty}^{\infty} f(x)\log(f(x))\,dx = H(X).
\end{aligned}$$

Bij $(*)$ hebben we toegepast dat de relatieve entropie $D(Y,X) \geq 0$ is, en bij $(**)$ dat X en Y kansverdelingen met dezelfde verwachtingswaarde en variantie hebben, dus dat $\int_{-\infty}^{\infty} f(x)\,dx = \int_{-\infty}^{\infty} g(x)\,dx = 1$, $\int_{-\infty}^{\infty} xf(x)\,dx = \int_{-\infty}^{\infty} xg(x)\,dx = 0$ en $\int_{-\infty}^{\infty} x^2 f(x)\,dx = \int_{-\infty}^{\infty} x^2 g(x)\,dx = \sigma^2$.

Als voorbeelden vergelijken we de entropie van een normale verdeling met variantie σ^2 met de entropie van een uniforme verdeling met dezelfde variantie.

Entropie van de normale verdeling

Zij X een normaal verdeelde stochast met verwachtingswaarde μ en variantie σ^2, dan heeft X de dichtheidsfunctie $f(x) = \frac{1}{\sqrt{2\pi}\,\sigma}\,e^{-\frac{(x-\mu)^2}{2\sigma^2}}$. Voor de entropie van X geldt

$$\begin{aligned}
H(X) &= -\int_{-\infty}^{\infty} f(x)\ {}^2\log(f(x))\,dx \\
&= -\int_{-\infty}^{\infty} f(x)({}^2\log(\frac{1}{\sqrt{2\pi}\,\sigma}) + {}^2\log(e^{-\frac{(x-\mu)^2}{2\sigma^2}}))\,dx \\
&= -{}^2\log(\frac{1}{\sqrt{2\pi}\,\sigma})\underbrace{\int_{-\infty}^{\infty} f(x)\,dx}_{=1} - \frac{1}{\log(2)}\int_{-\infty}^{\infty} f(x)\log(e^{-\frac{(x-\mu)^2}{2\sigma^2}})\,dx \\
&= {}^2\log(\sqrt{2\pi}\,\sigma) - \frac{1}{\log(2)}\int_{-\infty}^{\infty} f(x)(-\frac{(x-\mu)^2}{2\sigma^2})\,dx \\
&= {}^2\log(\sqrt{2\pi}\,\sigma) + \frac{1}{\log(2)}\frac{1}{2\sigma^2}\underbrace{\int_{-\infty}^{\infty} f(x)(x-\mu)^2\,dx}_{=Var(X)=\sigma^2} \\
&= {}^2\log(\sqrt{2\pi}\,\sigma) + \frac{1}{2\log(2)} = {}^2\log(\sqrt{2\pi}\,\sigma) + {}^2\log(\sqrt{e}) \\
&= {}^2\log(\sqrt{2\pi e}\,\sigma)
\end{aligned}$$

Entropie van de uniforme verdeling

Zij X een stochast met uniforme verdeling op het interval $[-a,a]$, dus met dichtheidsfunctie $f(x)=\frac{1}{2a}$ voor $x\in[-a,a]$ en $f(x)=0$ voor $x\notin[-a,a]$. We moeten eerst de variantie van X berekenen, hiervoor geldt

$$Var(X)=\int_{-a}^{a}\frac{1}{2a}x^2\,dx=\frac{1}{2a}\cdot\frac{x^3}{3}\Big|_{-a}^{a}=\frac{1}{2a}\,\frac{2a^3}{3}=\frac{a^2}{3}.$$

De variantie is dus voor $a=\sqrt{3}\,\sigma$ gelijk aan σ^2.

Voor de entropie geldt nu

$$H(X)=-\int_{-a}^{a}\frac{1}{2a}\ {}^2\log(\frac{1}{2a})\,dx=-\ {}^2\log(\frac{1}{2a})=\ {}^2\log(2a).$$

Voor een uniforme stochast met variantie σ^2, dus met $a=\sqrt{3}\,\sigma$, krijgen we dus de entropie

$$H(X)=\ {}^2\log(\sqrt{12}\,\sigma).$$

Omdat $\sqrt{12}\approx 3.464<4.132\approx\sqrt{2\pi e}$ is de entropie bij een uniforme verdeling inderdaad kleiner dan bij een normale verdeling met dezelfde variantie.

11.3 Voorwaardelijke entropie

Een belangrijke vraag is hoe zich de entropie van verschillende stochasten gedraagt als we deze combineren. We zouden verwachten, dat voor twee onafhankelijke stochasten X en Y de entropie van de combinatie van X en Y de som van de entropieën van X en Y is. Voor stochasten X, Y met uniforme verdelingen is dit juist eis (7) in onze lijst. Voor twee stochasten X en Y geldt inderdaad de stelling:

$H(X,Y)\le H(X)+H(Y)$ en
$H(X,Y)=H(X)+H(Y)$ alleen maar als X en Y onafhankelijk zijn.

Dit zien we als volgt in: We definiëren de kansen voor de stochasten als $p_i:=p(X=x_i)$ voor $1\le i\le n$, $y_j:=p(Y=y_j)$ voor $1\le j\le m$ en de gecombineerde kans als $r_{ij}:=p(X=x_i,Y=y_j)$. Als we voor vaste i de kansen r_{ij} voor alle j optellen, krijgen we de kans op x_i, dus geldt $p_i=\sum_{j=1}^{m}r_{ij}$ en evenzo $q_j=\sum_{i=1}^{n}r_{ij}$. We hebben dus

$$\begin{aligned}
H(X)+H(Y)&=-\sum_{i=1}^{n}p_i\ {}^2\log(p_i)-\sum_{j=1}^{m}q_j\ {}^2\log(q_j)\\
&=-\sum_{i=1}^{n}(\sum_{j=1}^{m}r_{ij})\ {}^2\log(p_i)-\sum_{j=1}^{m}(\sum_{i=1}^{n}r_{ij})\ {}^2\log(q_j)\\
&=-\sum_{i=1}^{n}\sum_{j=1}^{m}r_{ij}({}^2\log(p_i)+\ {}^2\log(q_j))=-\sum_{i=1}^{n}\sum_{j=1}^{m}r_{ij}\ {}^2\log(p_iq_j)\\
&\ge-\sum_{i=1}^{n}\sum_{j=1}^{m}r_{ij}\ {}^2\log(r_{ij})=H(X,Y).
\end{aligned}$$

De ongelijkheid $-\sum\sum r_{ij}\ {}^2\log(p_iq_j) \geq -\sum\sum r_{ij}\ {}^2\log(r_{ij})$ volgt hierbij weer uit de eigenschap (II) die we boven hebben bewezen, omdat ook p_iq_j een kansverdeling op $\{1,\ldots,n\}\times\{1,\ldots,m\}$ is.

We zien dat $H(X)+H(Y)$ alleen maar geldt als $p_iq_j = r_{ij}$ voor alle paren (i,j), dus als $p(X=x_i)\cdot p(Y=y_j) = p(X=x_i, Y=y_j)$, maar dit is precies de uitspraak dat X en Y onafhankelijk zijn.

Als we stochasten combineren, moeten we het natuurlijk ook over voorwaardelijke kansen hebben. Maar voorwaardelijke kansen zijn ook gewoon kansverdelingen: Als we de kans op een uitkomst x_i voor de stochast X onder de voorwaarde A weer als $p_i := p(X=x_i \mid A)$ beschrijven, is $P=(p_1,\ldots,p_n)$ een kansverdeling en $\sum_{i=1}^n p_i = 1$. We definiëren daarom de *voorwaardelijke entropie* $H(X \mid A)$ door

$$H(X \mid A) := -\sum_{i=1}^{n} p(X=x_i \mid A)\ {}^2\log(p(X=x_i \mid A)).$$

Nog algemener kunnen we ook de voorwaardelijke entropie van een stochast X, gegeven een andere stochast Y definiëren. Het idee hierbij is, dat de uitkomsten van de stochast Y de onzekerheid over de stochast X kunnen veranderen. We lopen dus over alle mogelijke uitkomsten y_j van de stochast Y, berekenen voor deze uitkomsten de voorwaardelijke entropie $H(X \mid y_j)$ en tellen deze entropieën op, met de kansen op de enkele y_j als gewichten.

Definitie: De *voorwaardelijke entropie* van de stochast X onder de voorwaarde van de stochast Y is gedefinieerd door

$$\begin{aligned} H(X \mid Y) &:= \sum_{j=1}^{m} H(X \mid Y=y_j)p(Y=y_j) \\ &= -\sum_{j=1}^{m}\sum_{i=1}^{n} p(X=x_i \mid Y=y_j)\ {}^2\log(p(X=x_i \mid Y=y_j))\cdot p(Y=y_j). \end{aligned}$$

Dat deze definitie enigszins zinvol is, zien we aan de twee extreme gevallen $Y=X$ en X,Y onafhankelijk:

(1) Als $Y=X$ is, dan is $p(X=x_i \mid X=x_j) = 1$ als $i=j$ en 0 als $i\neq j$. Maar dan geldt

$$\begin{aligned} H(X|X) &= -\sum_{j=1}^{n}\sum_{i=1}^{n} p(X{=}x_i|X{=}x_j)\ {}^2\log(p(X{=}x_i|X{=}x_j))p(X{=}x_j) \\ &= -\sum_{i=1}^{n} 1\cdot 0\cdot p(X=x_i) = 0. \end{aligned}$$

Er geldt dus

$$H(X \mid X) = 0.$$

Dit zegt dat er geen onzekerheid over X meer bestaat, als we de uitkomsten van X al kennen.

(2) Als X en Y onafhankelijk zijn, dan geldt $p(X = x_i \mid Y = y_j) = p(X = x_i)$, en hieruit volgt

$$\begin{aligned} H(X \mid Y) &= -\sum_{j=1}^{m}\sum_{i=1}^{n} p(X = x_i)\ {}^2\log(p(X = x_i))p(Y = y_j) \\ &= -\sum_{i=1}^{n} p(X = x_i)\ {}^2\log(p(X = x_i)) = H(X). \end{aligned}$$

Voor onafhankelijke stochasten X en Y geldt dus

$$H(X \mid Y) = H(X).$$

Dit betekent, dat de kennis over Y de onzekerheid bij X niet reduceert, en dat is precies wat we bij onafhankelijke stochasten zouden verwachten.

We kunnen nu ook de precieze samenhang tussen de voorwaardelijke entropie $H(X \mid Y)$ en de entropie van de combinatie van X en Y aangeven, er geldt namelijk

$$H(X,Y) = H(Y) + H(X \mid Y) \quad \text{of te wel} \quad H(X \mid Y) = H(X,Y) - H(Y).$$

Dit zien we als volgt in: We schrijven weer $r_{ij} := p(X = x_i, Y = y_j)$ voor de gecombineerde kans op x_i en y_j. Volgens de definitie van de voorwaardelijke kans geldt dat $p(X = x_i \mid Y = y_j) = \frac{r_{ij}}{q_j}$ en dus $r_{ij} = p(X = x_i \mid Y = y_j)q_j$, waarbij we weer $q_j := p(Y = y_j)$ schrijven. Er geldt dus:

$$\begin{aligned} H(X,Y) &= -\sum_{i,j} r_{ij}\ {}^2\log(r_{ij}) = -\sum_{i,j} r_{ij}\ {}^2\log(p(X = x_i \mid Y = y_j)q_j) \\ &= -\sum_{i,j} r_{ij}\ {}^2\log(p(X = x_i \mid Y = y_j)) - \sum_{i,j} r_{ij}\ {}^2\log(q_j) \\ &= -\sum_{i,j} r_{ij}\ {}^2\log(p(X = x_i \mid Y = y_j)) - \sum_{j=1}^{m} q_j\ {}^2\log(q_j) \\ &= -\sum_{i,j} p(X = x_i \mid Y = y_j)q_j\ {}^2\log(p(X = x_i \mid Y = y_j)) - H(Y) \\ &= H(X \mid Y) + H(Y). \end{aligned}$$

Hieruit volgt in het bijzonder dat

$$H(X \mid Y) \leq H(X),$$

want $H(X \mid Y) = H(X,Y) - H(Y) \leq H(X) + H(Y) - H(Y) = H(X)$, en dus is de voorwaardelijke entropie van een stochast nooit groter dan zijn absolute entropie. Ook dit is een eigenschap die we van een redelijke maat voor onzekerheid hadden kunnen verwachten, want door aanvullende informatie zouden we niet onzekerder over de uitkomsten van X worden.

11.4 Informatie

We hebben bij de voorwaardelijke entropie gezien, dat kennis over een stochast Y de onzekerheid over de stochast X kan reduceren. Het verschil van de entropieën $H(X) - H(X \mid Y)$ kunnen we dus zien als de informatie die Y aan onze kennis over X bijdraagt. Dit lijdt tot een precieze definitie van het begrip *informatie*, die we nu gaan behandelen.

Net als bij de entropie stellen we ook bij de informatie eisen aan een functie die de informatie van een gebeurtenis beschrijft. We schrijven $I(X = x_i)$ voor de informatie die de uitkomst x_i van de stochast X oplevert. Maar eigenlijk mag een abstracte definitie van informatie niet van de specifieke uitkomst afhangen, maar alleen maar van de kans op deze uitkomst. Dit geeft aanleiding tot de eerste eis die we aan een functie voor de informatie hebben:

(1) Er geldt $I(X = x_i) = I(p_i)$ voor $p_i = p(X = x_i)$.

Verder bekijken we de informatie van onafhankelijke gebeurtenissen: Als X en Y onafhankelijke stochasten zijn, geldt met $p_i = p(X = x_i)$ en $q_j = p(Y = y_j)$ dat $p(X = x_i, Y = y_j) = p_i q_j$. Maar het ligt voor de hand dat de informatie die in de uitkomst $X = x_i$ en $Y = y_j$ zit, de som van de informaties van de enkele uitkomsten is. Dit geeft de eis:

(2) Voor *onafhankelijke* stochasten X en Y met $p_i = p(X = x_i)$ en $q_j = p(Y = y_j)$ geldt $I(p_i q_j) = I(p_i) + I(q_j)$.

Met een soortgelijke (maar eenvoudigere) redenering als bij de entropie kan men nu aantonen dat de functie I noodzakelijk van de vorm $I(p) = -\lambda \log(p)$ is, en ook hier kiest men voor de logaritme met basis 2, dus definieert men:

Definitie: Voor een stochast X is de *informatie* van de uitkomst $X = x$ met $p(X = x) = p$ gegeven door

$$I(p) := -\,{}^2\log(p).$$

> Deze definitie van informatie is in ieder geval ook in overeenstemming met onze intuïtie dat het optreden van een gebeurtenis met een kleine kans meer informatie oplevert dan een gebeurtenis met een grote kans, dus van *het gewone*.

Een belangrijke rechtvaardiging van deze definitie van informatie vinden we weer in de communicatietheorie: Als we een bit-string van lengte n produceren door toevallig n keer een 0 of 1 te kiezen, heeft elke bit van de string de informatie $I(\frac{1}{2}) = -\,{}^2\log(\frac{1}{2}) = {}^2\log(2) = 1$ en de totale informatie in de string is dus $-n\ {}^2\log(\frac{1}{n}) = n$, omdat de keuzes van de bits onafhankelijk zijn. Het is daarom ook gebruikelijk, informatie (en entropie) in *bits* aan te geven.

Verband tussen informatie en entropie

Met behulp van het begrip van informatie kunnen we nu de entropie herinterpreteren. Er geldt

$$H(X) = -\sum p_i \, {}^2\log(p_i) = \sum p_i \cdot I(p_i)$$

dus is de entropie het gemiddelde van de informatie in de enkele uitkomsten, gewogen met de kansen van de uitkomsten. Maar in de taal van de kansrekening is dit gewogen gemiddelde juist de verwachtingswaarde:

Merk op: De entropie $H(X)$ van een stochast X is de verwachtingswaarde van de informatie van de enkele uitkomsten van de stochast.

Dit kunnen we ook nog iets anders formuleren: Een uitkomst met informatie $I = {}^2\log(n)$ heeft kans $p = \frac{1}{n}$. Als de uitkomst bij een uniforme verdeling hoort, is $\frac{1}{p} = n$ het aantal mogelijke uitkomsten. Dit betekent dat we voor een uniforme verdeling het aantal mogelijke uitkomsten kunnen schrijven als $n = 2^I$, waarbij I de informatie is die in een enkele uitkomst zit. Maar we hebben net gezien dat de entropie de verwachtingswaarde van de informatie in de enkele uitkomsten is, dus kunnen we $2^{H(X)}$ interpreteren als het gemiddelde aantal alternatieven, dat we bij de stochast X kunnen verwachten. Dit kunnen we ook als volgt formuleren:

Merk op: De onzekerheid bij een stochast X is even groot als de onzekerheid bij een uniforme verdeling met $2^{H(X)}$ mogelijke uitkomsten. Anders gezegd is $2^{H(X)}$ het gemiddelde aantal alternatieven, dat we bij een kansexperiment voor de stochast X verwachten.

We hebben in het begin van deze sectie gesteld, dat het verschil van de entropieën $H(X) - H(X \mid Y)$ de informatie is, die Y over X onthult. Als notatie hiervoor gebruiken we

$$I(X \mid Y) := H(X) - H(X \mid Y).$$

Er geldt $I(X \mid X) = H(X)$, want $H(X \mid X) = 0$, en dit is ook zinvol omdat kennis van X de onzekerheid over X precies moet compenseren. Aan de andere kant geldt voor onafhankelijke stochasten X en Y dat $I(X \mid Y) = 0$, want $H(X \mid Y) = H(X) + H(Y)$. Ook dit is juist wat we nodig hebben, want onafhankelijke stochasten mogen onderling geen informatie onthullen.

Bij de definitie van $I(X \mid Y)$ kijken we naar de gemiddelde reductie die de enkele uitkomsten van Y voor de entropie van X opleveren. We kunnen natuurlijk ook naar de informatie kijken, die een bepaalde uitkomst $Y = y$ voor de stochast Y over X oplevert, deze is gedefinieerd door

$$I(X \mid Y = y) = H(X) - H(X \mid Y = y).$$

Er bestaat een iets verrassende symmetrie voor het onthullen van informatie van een stochast over de andere. We hebben namelijk

$$\begin{aligned} I(X \mid Y) &= H(X) - H(X \mid Y) = H(X) - (H(X,Y) - H(Y)) \\ &= H(Y) + (H(X) - H(X,Y)) = H(Y) - H(Y \mid X) \\ &= I(Y \mid X), \end{aligned}$$

dus onthult de stochast X net zo veel informatie over Y als de stochast Y over X onthult.

11.5 Toepassing: Automatische Taalherkenning

Als voorbeeld voor de toepassing van de concepten van entropie en informatie bekijken we het probleem van de automatische taalherkenning op geschreven tekst. Voor een mens is dit meestal nauwelijks een probleem, tenminste bij bekende talen of bij talen waar men iets over weet, maar de automatisering hiervan is al een stukje lastiger.

Onze aanpak is, de relatieve frequenties van de letters te gebruiken. Het is natuurlijk bekend dat de letters in het alfabet niet even vaak gebruikt worden, in het Nederlands is bijvoorbeeld de letter `E` de meest frequente. Het idee is dat de relatieve frequenties voor verschillende talen er verschillend uit zien en dat we hiermee de talen kunnen onderscheiden.

> Vanaf de 16de eeuw zijn de relatieve frequenties in de cryptanalyse gebruikt om versleutelingen met monoalfabetische substitutie (elke letter wordt door een andere letter vervangen, maar één letter steeds door dezelfde) te kraken. Tot op die tijd dacht men eigenlijk dat zo'n versleuteling niet te kraken was, omdat er veel te veel sleutels bestaan ($26! \approx 4.03 \cdot 10^{26}$) om alle te proberen. Maar als men al weet dat de meest frequente letter in de versleuteling een `E` is en de volgende waarschijnlijk een `N` kan men al gauw verdere letters gokken.
>
> Het idee dat de letters überhaupt verschillende frequenties hebben, is waarschijnlijk pas na de opkomst van de boekdrukkerij (door Gutenberg) ontdekt, omdat de loodletters verschillend snel versleten waren.

Voor een gegeven taal kan men op een grote achtergrondtekst de frequenties tellen en dit als kansverdeling van de stochast X die de letters beschrijft nemen. Men krijgt zo de kansen $p_1 := p(X = \texttt{A})$, $p_2 := p(X = \texttt{B})$, $\ldots$, $p_{26} := p(X = \texttt{Z})$, $p_{27} := p(X = \textit{spatie})$.

Tabel III.1 geeft deze kansverdelingen voor de vier talen *Nederlands*, *Engels*, *Duits* en *Fins* weer. De gebruikte achtergrondtekst is een tekst van de Europese Unie die in de verschillende talen vertaald is en ongeveer 50000 letters bevat. Uit deze tabel kan men concluderen dat de kansverdelingen voor Nederlands, Engels en Duits enigszins op elkaar lijken, terwijl de verdeling voor Fins er behoorlijk anders uit ziet. Bijvoorbeeld bepaalt de relatieve frequentie van de *spatie* de gemiddelde lengte van de woorden (namelijk door $l_{gem} = \frac{1}{p} - 1$) en men ziet dat de woorden in het Fins gemiddeld duidelijk langer zijn dan in de andere talen.

Een betere voorstelling van de frequentieverdelingen dan met de tabel krijgt men door de verdelingen als histogrammen te plotten, zo als in Figuur III.2 te zien. Hier valt bijvoorbeeld op, dat er in het Fins meer letters met een relatief hoge frequentie zijn, en dat in het Nederlands en Duits de letter `E` met duidelijke afstand de hoogste frequentie heeft.

letter	Nederlands	Engels	Duits	Fins
A	5.55%	6.37%	4.14%	9.57%
B	1.45%	0.99%	1.82%	0.10%
C	1.45%	3.20%	2.09%	0.05%
D	4.72%	2.56%	4.09%	1.40%
E	17.31%	9.93%	13.89%	8.50%
F	0.68%	1.95%	2.28%	0.07%
G	2.79%	1.41%	2.67%	0.19%
H	1.83%	3.00%	3.00%	1.77%
I	6.09%	7.62%	8.22%	9.90%
J	0.70%	0.10%	0.14%	1.57%
K	1.51%	0.27%	1.21%	4.74%
L	2.87%	2.93%	2.83%	3.75%
M	1.98%	2.52%	2.81%	2.65%
N	8.67%	7.63%	9.14%	8.08%
O	4.94%	7.73%	2.92%	6.68%
P	1.53%	2.78%	1.03%	1.78%
Q	0.01%	0.04%	0.01%	0.01%
R	5.81%	5.15%	6.69%	2.16%
S	3.44%	4.92%	5.10%	8.24%
T	5.63%	8.30%	5.40%	9.54%
U	2.01%	2.57%	3.85%	4.70%
V	2.77%	0.70%	0.80%	2.10%
W	0.67%	0.75%	0.77%	0.02%
X	0.05%	0.12%	0.05%	0.01%
Y	0.04%	0.84%	0.06%	1.71%
Z	0.55%	0.02%	1.36%	0.05%
spatie	14.94%	15.61%	13.63%	10.64%

Tabel III.1: Letter frequenties voor vier verschillende talen

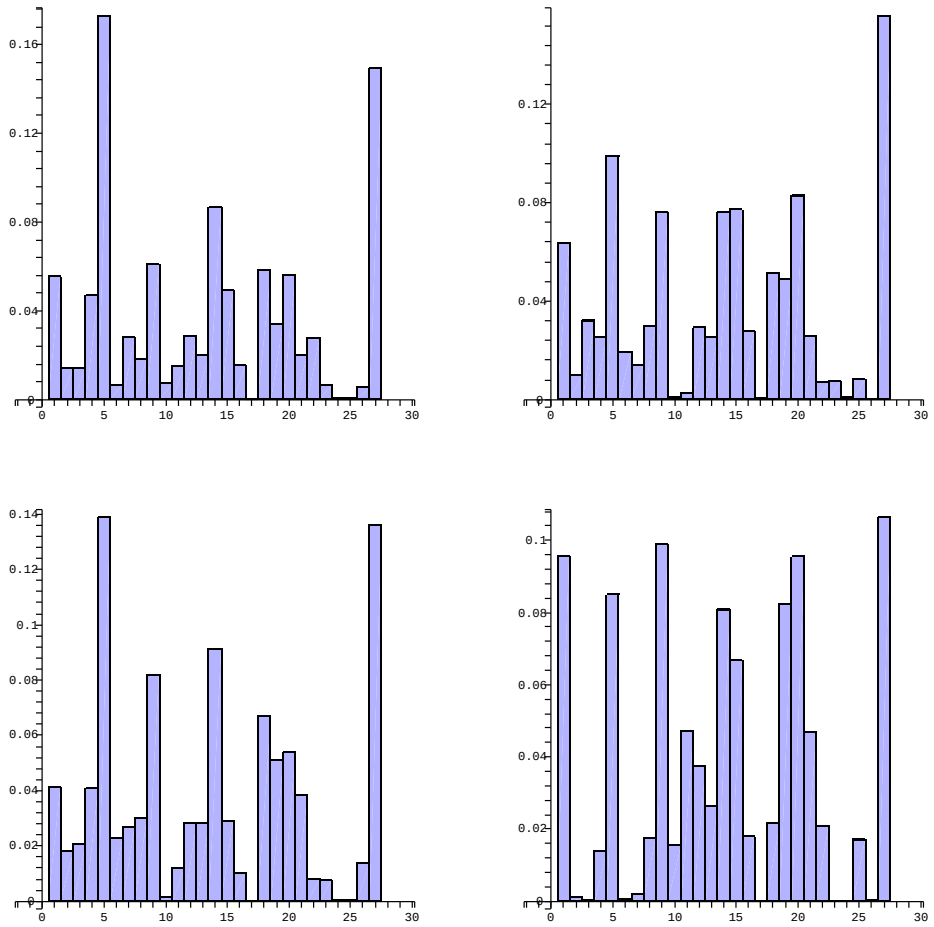
Als we de frequentieverdelingen als kansverdelingen opvatten, kunnen we voor de verschillende talen de entropieën van deze verdelingen uitrekenen, dit geeft de volgende waarden:

$$H(\text{Nederlands}) = 4.019, \qquad H(\text{Engels}) = 4.070,$$
$$H(\text{Duits}) = 4.109, \qquad H(\text{Fins}) = 3.982.$$

Met de interpretatie van de entropie met behulp van informatie geeft dit:

$$2^{H(\text{Nederlands})} = 16.21, \qquad 2^{H(\text{Engels})} = 16.80,$$
$$2^{H(\text{Duits})} = 17.26, \qquad 2^{H(\text{Fins})} = 15.80.$$

Het gemiddelde aantal alternatieven, dat we in de verschillende talen voor een letter verwachten, ligt dus tussen 15.80 voor Fins en 17.26 voor Duits, terwijl we bij een uniforme verdeling 27 alternatieven zouden hebben.



Figuur III.2: Letter-frequentieverdelingen voor Nederlands (links boven) en Engels (rechts boven), Duits (links onder) en Fins (rechts onder).

Classificatie van patronen

Een typisch probleem in de patroonherkenning is, gegeven een aantal klassen $K_1, \ldots, K_n$ van mogelijke patronen, een nieuw patroon aan een van de klassen K_i toe te wijzen. Denk bij de klassen bijvoorbeeld aan letters in de handschriftherkenning, aan woorden of fonemen in de spraakherkenning of objecten in de beeldherkenning. In ons voorbeeld van de automatische taalherkenning zijn de klassen natuurlijk de talen en het nieuwe patroon is een nieuwe tekst.

In het verleden is geprobeerd, regels te vinden waarmee de klasse van een nieuw patroon bepaald kan worden. Maar er is gebleken dat dit slechts zeer beperkt inzetbaar is en de beste methoden in de patroonherkenning gebruiken nu probabilistische modellen, bijvoorbeeld (hidden) Markov modellen of/en neuronale netwerken.

Er zijn verschillende mogelijkheden voor de rol die kansverdelingen bij het classificeren van patronen kunnen spelen:

- Het nieuwe patroon wordt door een vector (of een rij vectoren) in de *kenmerkruimte* (feature space) weergegeven. De klassen zijn gerepresenteerd door kansverdelingen op de kenmerkruimte die aangeven hoe groot de kans is dat een patroon met een zekere vector bij deze klasse hoort. Het patroon wordt dan aan de klasse toegewezen waarvoor deze kans maximaal is.

- Ook voor het patroon wordt een kansverdeling bepaald en er wordt de klasse gekozen, waarvoor deze kansverdeling het meeste op de eerder berekende kansverdeling van de klasse lijkt.

We zullen de tweede insteek nu eens nader bekijken, omdat die minder voor de hand liggend lijkt als de eerste. In het voorbeeld van de automatische taalherkenning zijn de kansverdelingen gegeven door de relatieve frequenties van de letters. Voor een nieuwe tekst waarvan we de taal willen bepalen moeten we daarom ook de frequentieverdeling berekenen en vervolgens deze kansverdeling met de bekende kansverdelingen van de verschillende talen vergelijken. De aanname is dan, dat de tekst bij die taal hoort waarvoor de kansverdelingen het meeste op elkaar lijken.

De vraag is nu hoe men objectief bepaald, dat een kansverdeling meer op een dan op een andere lijkt.

Afstanden tussen kansverdelingen

Om een eenvoudige notatie te krijgen, beschrijven we een discrete kansverdeling P op de verzameling $\Omega = \{1, \ldots, n\}$ door de vector van kansen $p_i := p(i)$, dus $P = (p_1, p_2, \ldots, p_n)$. Voor een tweede kansverdeling $Q = (q_1, q_2, \ldots, q_n)$ op dezelfde verzameling Ω willen we nu een afstand tussen P en Q definiëren.

Een voor de hand liggende idee is, de Euclidische afstand van de vectoren P en Q in de n-dimensionale ruimte te nemen, dit geeft

$$d_2(P,Q) = \left(\sum_{i=1}^{n} (p_i - q_i)^2 \right)^{\frac{1}{2}}.$$

Maar net zo goed zouden we in plaats van de kwadraten van de verschillen tussen p_i en q_i ook de absolute waarden van de verschillen kunnen optellen:

$$d_1(P,Q) = \sum_{i=1}^{n} |p_i - q_i|.$$

We kunnen zelfs heel algemeen een macht van de verschillen tussen p_i en q_i optellen, dit geeft

$$d_r(P,Q) = \left(\sum_{i=1}^{n} |p_i - q_i|^r \right)^{\frac{1}{r}}.$$

Hierbij hoeft r niet eens een geheel getal te zijn, we kunnen een willekeurige r met $0 < r < \infty$ kiezen. De reden dat we bij een r-de macht ook weer een

r-de machtswortel trekken, heeft ermee te maken dat men graag wil dat een vermenigvuldiging van de vectoren met een constante factor tot een vermenigvuldiging van de afstand met dezelfde factor leidt.

Voor de volledigheid noemen we nog een verdere afstand, die we formeel kunnen krijgen als we bij $d_r(P,Q)$ de $r \to \infty$ laten lopen. Dan krijgen we namelijk de afstand

$$d_\infty(P,Q) = \max_i |p_i - q_i|$$

die gewoon het grootste verschil in een van de componenten aangeeft. Maar als we naar vectoren van kansverdelingen kijken, is dit meestal geen bijzonder nuttige afstand.

De vraag welke afstand nu een slimme keuze is, heeft helaas geen eenvoudig antwoord. Het hangt namelijk van het probleem af. Hoe groter de waarde van de parameter r is hoe groter is relatief het gewicht van de grotere verschillen en hoe kleiner de invloed van kleine verschillen. Als r heel groot wordt, speelt inderdaad alleen maar het grootste verschil nog een rol. In sommige problemen is het misschien wenselijk, kleine verschillen te onderdrukken, maar soms ligt de informatie juist in de componenten met kleine verschillen.

In een iets algemenere opzet zou men voor elke component een functie $d_i(p_i, q_i)$ definiëren, die de afstand in deze component aangeeft. Als afstand krijgt men dan

$$d(P,Q) = \sum_{i=1}^{n} d_i(p_i, q_i).$$

Hierbij kan de functie d_i aan de ene kant ervoor zorgen, dat componenten met belangrijkere informatie een hoog gewicht krijgen, maar ook dat afhankelijk van de kansen een hoger of lager gewicht toegewezen wordt.

Een eenvoudig voorbeeld hiervan is het toewijzen van gewichten aan de enkele componenten, dus bijvoorbeeld

$$d(P,Q) = \sum_{i=1}^{n} w_i|p_i - q_i| \quad \text{of} \quad d(P,Q) = \sum_{i=1}^{n} w_i p_i q_i.$$

Het laatste is een inproduct van de twee vectoren P en Q en geeft weer dat we in principe ook de hoek tussen twee vectoren als een soort afstand kunnen interpreteren, zeker als de lengte van de vectoren genormeerd is.

> Het idee de afstand tussen kansverdelingen met behulp van een inproduct te berekenen wordt bijvoorbeeld in (eenvoudige) zoekmachines gebruikt, de gewichten zijn dan bijvoorbeeld de negatieve logaritmen van de relatieve frequenties van de woorden. Zo houdt men rekening ermee, dat frequente woorden weinig informatie over een document geven, terwijl minder frequente woorden vaak een belangrijke hint zijn.

De afstanden die we tot nu toe hebben bekeken, hebben op zich weinig met kansverdelingen te maken, want we hebben eigenlijk alleen maar naar vectoren gekeken. Het enige wat van de kansverdelingen over blijft, is dat de som van de componenten 1 is, dus dat $\sum_{i=1}^{n} p_i = 1$.

Kullback-Leibler afstand

Maar natuurlijk hebben we eerder in deze les ook al een maat voor de afstand tussen kansverdelingen gezien, namelijk de Kullback-Leibler afstand (of relatieve entropie).

We hadden gezien dat de Kullback-Leibler afstand $D(P,Q)$ het verschil tussen $-\sum p_i \ {}^2\log(q_i)$ en de entropie $H(P)$ van de kansverdeling P aangeeft, dus dat

$$D(P,Q) = \sum_{i=1}^{n} p_i \ {}^2\log(\frac{p_i}{q_i}) = (-\sum_{i=1}^{n} p_i \ {}^2\log(q_i)) - H(P).$$

Als we nu $2^{H(P)}$ als het gemiddelde aantal alternatieven interpreteren, die we bij een stochast X met kansverdeling P verwachten, kunnen we ook de Kullback-Leibler afstand herinterpreteren: Er geldt

$$2^{H(P)+D(P,Q)} = 2^{H(P)} \cdot 2^{D(P,Q)},$$

dus is $2^{D(P,Q)}$ de factor waarmee we het gemiddelde aantal alternatieven moeten vermenigvuldigen, omdat we de *verkeerde* kansverdeling Q in plaats van P veronderstellen.

De volgende tabellen geven links de Kullback-Leibler afstanden tussen de talen uit het voorbeeld met de frequentieverdelingen en rechts de factoren $2^{D(P,Q)}$. Hierbij betekent bijvoorbeeld een factor 1.138 een afwijking van 13.8% van het aantal verwachte alternatieven bij de juiste kansverdeling. Merk op dat de tabellen niet symmetrisch zijn, omdat we de gewone Kullback-Leibler afstand $D(P,Q)$ en niet de symmetrische versie $d_{KL}(P,Q)$ toepassen.

taal	NL	EN	DU	FI
NL	-	0.186	0.091	0.471
EN	0.171	-	0.155	0.458
DU	0.090	0.177	-	0.610
FI	0.397	0.373	0.453	-

taal	NL	EN	DU	FI
NL	-	1.138	1.065	1.386
EN	1.126	-	1.114	1.373
DU	1.064	1.130	-	1.527
FI	1.317	1.295	1.368	-

Het is opvallend hoe sterk Duits en Fins van elkaar afwijken, terwijl Nederlands en Duits redelijk dicht bij elkaar liggen.

> De Kullback-Leibler afstand speelt een belangrijke rol bij het bepalen van de parameters van probabilistische modellen. Het idee is dat op een zekere hoeveelheid training materiaal de kansen p_i worden bepaald en vervolgens een probabilistisch model gebouwd wordt, dat van enkele parameters afhangt. Dit kan bijvoorbeeld een normale verdeling zijn, met als parameters de verwachtingswaarde en de variantie. Deze parameters kunnen meestal niet rechtstreeks berekent worden, maar worden in een iteratief proces benadert, waarbij de Kullback-Leibler afstand stapsgewijs kleiner wordt. Als geen verbetering meer bereikt wordt, worden deze parameters voor het model gekozen.

Belangrijke begrippen in deze les

- onzekerheid, entropie
- relatieve entropie, Kullback-Leibler afstand
- entropie bij continue kansverdelingen
- maximale entropie bij normale verdeling
- voorwaardelijke entropie
- informatie
- afstanden tussen kansverdelingen

Opgaven

84. Er vinden twee paardenraces plaats, het eerste met 7 paarden en het tweede met 8 paarden. In de eerste race hebben 3 paarden kans $\frac{1}{6}$ om te winnen, de andere 4 hebben kans $\frac{1}{8}$. In de tweede race hebben 2 paarden kans $\frac{1}{4}$ om te winnen en de andere 6 kans $\frac{1}{12}$. Maak eerst een gok in welk van de races de uitkomst onzekerder is (en geef een reden hiervoor), en bereken dan de entropieën voor de twee races.

85. Er wordt met een eerlijke dobbelsteen gedobbeld. De stochast X geeft het aantal ogen dat gedobbeld wordt, de stochast Y heeft de waarde 0 of 1, afhankelijk of het aantal ogen even of oneven is. Bereken $H(X)$, $H(Y)$ en $H(X \mid Y)$.

86. Voor een geheel getal N neemt de stochast X volgens een uniforme verdeling de waarden $1, 2, \ldots, 2N$ aan. De stochast Y is 0 als de waarde van X even is en Y is 1 als de waarde van X oneven is. Laat zien dat $H(X \mid Y) = H(X) - 1$ en dat $H(Y \mid X) = 0$.

87. De uitkomsten van twee (eerlijke) dobbelstenen worden door de stochasten X en Y beschreven, de som van de twee dobbelstenen door de stochast Z. Ga na dat voor de combinatie van de stochasten X en Y geldt dat $H(X, Y) = H(X) + H(Y)$ en dat $H(Z) < H(X, Y)$.

88. Een stochast X heeft een binomiale verdeling met parameters n en p, d.w.z. de kans op de i-de uitkomst is $p(X = i) = \binom{n}{i} p^i (1-p)^{n-i}$. Laat zien dat

$$H(X) = -n(p\ {}^2\log(p) + (1-p)\ {}^2\log(1-p)).$$

89. Laat zien dat de entropie $H(X)$ van een continue stochast X met een exponentiële verdeling met dichtheidsfunctie

$$f(x) = \frac{1}{\lambda} e^{-\frac{x}{\lambda}} \text{ voor } x \geq 0$$

gegeven is door

$$H(X) = {}^2\log(\lambda\, e).$$

90. Bij een *best-of-five* tennis match is de speler de winnaar die als eerste drie sets gewonnen heeft. Stel dat de spelers A en B (ongeveer) even sterk zijn, zo dat een set met kans $\frac{1}{2}$ door A of B gewonnen wordt.

    Zij X de stochast die de mogelijke rijtjes van gewonnen sets beschrijft, dus bijvoorbeeld AAA, $ABBAA$ of $ABBB$. Verder zij Y de stochast die het aantal benodigde sets aangeeft (en dus een van de waarden 3, 4 of 5 heeft).

    Bepaal de entropieën $H(X)$ en $H(Y)$ en de voorwaardelijke entropieën $H(Y \mid X)$ en $H(X \mid Y)$.

91. Waar zit meer informatie in, in een string van 10 letters uit $\{A, \ldots, Z\}$ of in een string van 26 cijfers uit $\{0, \ldots, 9\}$?

92. Er wordt met een eerlijke dobbelsteen gedobbeld. Wat is de informatie, die de kennis dat het aantal ogen niet door 3 deelbaar is, over het aantal ogen onthult?

93. Uit onderzoek is gebleken dat 70% van de mannen donker haar hebben en 25% van de vrouwen blond zijn. Verder is bekend dat 80% van de blonde vrouwen met een donkerharig man trouwen. Hoeveel informatie over de haarkleur van de man onthult de haarkleur van zijn vrouw?

Les 12 Markov processen en Markov modellen

We hebben in het kader van de Lineaire Algebra in Wiskunde 1 een aantal voorbeelden van systemen gezien, die zich door overgangsmatrices laten beschrijven. Voorbeelden hiervan waren:

- Populaties die zich volgens overgangen tussen de verschillende generaties ontwikkelen.
- De verspreiding van de Euro munten over de verschillende landen.

Iets algemener gesproken hebben we het hierbij over systemen, die gekarakteriseerd zijn door: (1) mogelijke *toestanden* van het systeem; en (2) *overgangen* tussen deze toestanden.

We zullen het in deze (en de volgende) les over dit soort systemen hebben, waarbij we vooral naar het belangrijke geval kijken, dat de overgangen door kansverdelingen worden beschreven.

12.1 Markov processen

Als de overgangen tussen de mogelijke toestanden van een systeem door kansverdelingen gegeven zijn, spreekt men meestal van *Markov processen*.

Definitie: Een *Markov proces* wordt door de volgende gegevens gekarakteriseerd:

- een aantal mogelijke toestanden $S_1, S_2, \ldots, S_N$, die we *states* noemen;
- op elk tijdstip $t = 0, 1, 2, \ldots$ een state $q_t \in \{S_1, \ldots, S_N\}$, waarin het systeem zich op dit tijdstip bevindt;
- gegeven de states $q_0, q_1, \ldots, q_{t-1}$ op de tijdstippen $0, 1, \ldots, t-1$, de kansverdeling dat het systeem op tijdstip t in de state S_j terecht komt, d.w.z. de voorwaardelijke kansen

$$p(q_t = S_j \mid q_{t-1} = S_{i_{t-1}}, \ldots, q_1 = S_{i_1}, q_0 = S_{i_0}).$$

Het probleem om in de praktijk een proces als een Markov proces te beschrijven, zit in de exponentiële groei van het aantal mogelijke states op de tijdstippen $0, 1, \ldots, t-1$, dit zijn er namelijk N^t. Voor elke van deze mogelijkheden moeten we in principe een aparte kansverdeling voor de states op tijdstip t aangeven, maar dat is natuurlijk voor grotere waarden van t ondoenlijk.

Als we nog eens naar het voorbeeld van de taalherkenning middels letterfrequenties kijken, kunnen we dit zien als een Markov proces waarbij de states de verschillende letters zijn. In dit geval zouden voor elk beginstuk van t letters een kansverdeling voor de daarop volgende letter moeten bepalen. Voor een beginstuk van 8 letters zijn dit bijvoorbeeld $27^8 = 282429536481$ verdelingen, en die kunnen we nog bepalen nog opslaan.

Maar dit voorbeeld wijst ook al een mogelijke oplossing aan: We kunnen ervan uitgaan dat de kansverdeling voor de 9-de letter niet erg verandert als

we de eerste letter q_0 veranderen en waarschijnlijk zal ook de letter op tijdstip $t = 1$ nog geen grote betekenis voor de kansen van de verschillende waarden van q_8 hebben.

Dit leidt tot het idee, de kansverdeling voor de states op tijdstip t te benaderen door de kansverdeling die alleen maar met de k voorafgaande states rekening houdt, d.w.z. we nemen aan dat

$$p(q_t = S_j | q_{t-1} = S_{i_{t-1}}, \ldots, q_0 = S_{i_0}) \approx p(q_t = S_j | q_{t-1} = S_{i_{t-1}}, \ldots, q_{t-k} = S_{i_{t-k}})$$

een voldoende nauwkeurige benadering geeft.

Definitie: Een Markov proces, waarbij de kans op de states op tijdstip t alleen maar van de k voorafgaande states afhangt, heet een *Markov proces van orde k*. Hierbij wordt verondersteld dat de kansen niet van het tijdstip t afhangen, maar alleen maar van de rij voorafgaande states.

Voor een systeem met N mogelijke states wordt een Markov proces van orde k dus beschreven door de N^k kansverdelingen

$$p(q_t = S_j \mid q_{t-1} = S_{i_{t-1}}, \ldots, q_{t-k} = S_{i_{t-k}})$$

waarbij $(S_{i_{t-1}}, \ldots, S_{i_{t-k}})$ over alle mogelijke combinaties van states op de tijdstippen $t-1, \ldots, t-k$ loopt (onafhankelijk van t).

Bij een **Markov proces van orde** 0 speelt de geschiedenis helemaal geen rol, de states worden alleen maar volgens een kansverdeling op de states voortgebracht. Zo'n Markov proces krijgen we bijvoorbeeld, als we (zo als in de laatste les) alleen maar de relatieve frequenties van de letters in een taal bepalen en vervolgens letters volgens deze kansverdeling produceren. De relatieve frequentie van de letters zal dan wel kloppen, maar bijvoorbeeld de relatieve frequenties van paren van letters niet meer. Hiervoor hebben we een Markov proces van orde 1 nodig.

Een **Markov proces van orde** 1 is gekarakteriseerd door de overgangskansen $a_{ij} := p(q_t = S_j \mid q_{t-1} = S_i)$. Omdat we veronderstellen dat deze kansen onafhankelijk van t zijn, kunnen we de kansen in een *overgangsmatrix* $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ invullen. Voor deze overgangsmatrix A geldt dat $a_{ij} \geq 0$ en dat $\sum_{j=1}^{n} a_{ij} = 1$ voor alle $i = 1, \ldots, n$, omdat het systeem vanuit state S_i naar een van de states S_j moet overgaan.

Een handige eigenschap van de overgangsmatrix A is dat we met behulp van de machten van A makkelijk kunnen berekenen wat er over een aantal stappen gebeurt: Het element (i, j) van A^k geeft de kans aan, dat het systeem in (precies) k stappen van state S_i naar S_j gaat.

Een Markov proces van orde 1 laat zich ook overzichtelijk door een *graaf* of *state diagram* representeren: De states zijn punten en de overgangen zijn pijltjes tussen de states, met de kans voor de overgangen als labels.

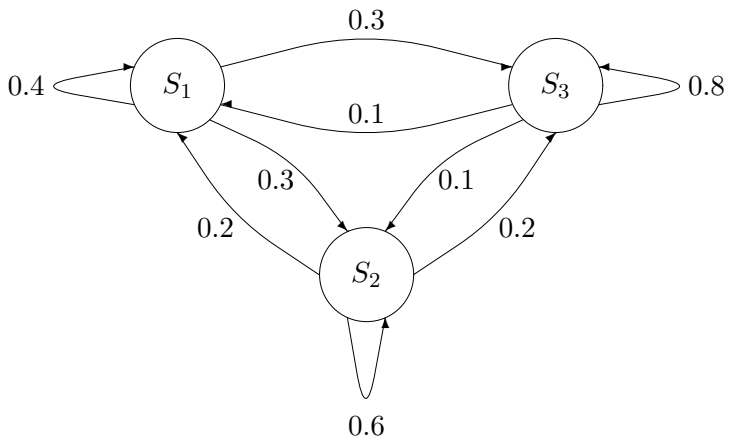
Als we bijvoorbeeld het weer als een (eenvoudige) Markov proces willen beschrijven, zouden we misschien de drie states

$$S_1 = \text{ regen}, \quad S_2 = \text{ bewolkt}, \quad S_3 = \text{ zonnig}$$

kunnen kiezen. Als overgangsmatrix veronderstellen we de (erg optimistische) matrix

$$A = \begin{pmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{pmatrix}.$$

Dan heeft deze Markov proces het state diagram uit Figuur III.3.



Figuur III.3: Markov proces van orde 1 voor het weer.

Aan de hand van de overgangskansen kunnen we een aantal vragen makkelijk beantwoorden:

(1) Wat is de kans op drie dagen zon gevolgd van een dag regen?

(2) Wat is de kans dat het weer precies d dagen hetzelfde blijft?

Bij vraag (1) willen we de kans op de rij $O = S_3S_3S_3S_1$ van states weten. Maar de overgang $S_3 \to S_3$ heeft kans $a_{33} = 0.8$ en de overgang $S_3 \to S_1$ heeft kans $a_{31} = 0.1$, dus is de kans op deze rij $0.8 \cdot 0.8 \cdot 0.1 = 0.064$. Hierbij veronderstellen we wel, dat de vraag op een dag gesteld wordt, waar het al zonnig is, dus waar we al in state S_3 zitten.

Vraag (2) gaat over een rij $O = \underbrace{S_iS_i \ldots S_i}_{d} S_j$ van states, waarbij er precies d keer de state S_i voorkomt en de state S_j verschillend is van S_i. Maar de kans dat we van state S_i naar een state verschillend van S_i gaan is $1 - a_{ii}$, dus is de kans $p(O)$ van deze rij states $p(O) = a_{ii}^{d-1} \cdot (1 - a_{ii})$.

We kunnen nu zelfs de verwachtingswaarde voor het aantal dagen d berekenen, die we in state S_i blijven, er geldt:

$$E[d] = \sum_{d=1}^{\infty} d \cdot a_{ii}^{d-1} \cdot (1 - a_{ii}) = \frac{1}{1 - a_{ii}}.$$

Dit zien we als volgt in: Voor de meetkundige reeks geldt $\sum_{d=0}^{\infty} x^d = \frac{1}{1-x}$ als $|x| < 1$. Maar $\sum_{d=1}^{\infty} dx^{d-1} = (\sum_{d=0}^{\infty} x^d)'$, omdat we in dit geval termsgewijs

mogen afleiden. Aan de andere kant is $(\frac{1}{1-x})' = \frac{1}{(1-x)^2}$, en dus is

$$\sum_{d=1}^{\infty} d \cdot a_{ii}^{d-1} \cdot (1 - a_{ii}) = \frac{1}{(1-a_{ii})^2}(1 - a_{ii}) = \frac{1}{1-a_{ii}}.$$

In ons optimistisch model van het weer is de kans dat het blijft regenen $a_{11} = 0.4$, dus is de verwachtingswaarde voor het aantal regendagen achter elkaar gelijk aan $\frac{1}{1-0.4} = \frac{1}{0.6} \approx 1.67$. Net zo krijgen we voor het verwachte aantal bewolkte dagen achter elkaar de waarde $\frac{1}{1-0.6} = 2.5$ en voor het aantal zonnige dagen achter elkaar hebben we de verwachtingswaarde $\frac{1}{1-0.8} = 5$.

12.2 Stochastische automaten

Bij de Markov processen zijn we ervan uitgegaan dat het systeem op tijdstippen $t = 0, 1, \ldots$ van een state naar een andere state overgaat. In sommige samenhangen wordt zo'n overgang veroorzaakt door een input aan het systeem. Maar dan is het plausibel dat de overgangskansen ook van de mogelijke inputs kunnen afhangen. Dit betekent, dat er bij een Markov proces van orde 1 voor elke mogelijke input een aparte overgangsmatrix is. Zo'n systeem noemt men ook een *stochastische automaat*.

We bekijken dit aan de hand van het voorbeeld van een *emotionele robot*. Stel de robot heeft drie mogelijke states, namelijk

$$S_1 = \text{ gelukkig}, \quad S_2 = \text{ bedroefd}, \quad S_3 = \text{ mal}$$

en er zijn de twee mogelijke inputs

$$X = \text{ 'hallo schat'} \quad \text{en} \quad Y = \text{ 'oude roestdoos'}$$

dan horen bij deze twee inputs misschien de overgangsmatrices

$$A_X = \begin{pmatrix} 0.7 & 0.2 & 0.1 \\ 1.0 & 0 & 0 \\ 0 & 0 & 1.0 \end{pmatrix} \quad \text{en} \quad A_Y = \begin{pmatrix} 0 & 0.9 & 0.1 \\ 0 & 0.6 & 0.4 \\ 0 & 0 & 1.0 \end{pmatrix}.$$

Een state zo als S_3 waaruit een systeem niet meer kan ontsnappen, heet een *absorberende* state.

Ook voor een stochastische automaat laten zich de kansen over langere perioden door producten van de overgangsmatrices berekenen, als de rij van inputs bekend is. Als de robot bijvoorbeeld op de werkdagen input Y maar op het weekend input X te horen krijgt, zijn de overgangskansen van maandag tot maandag gegeven door

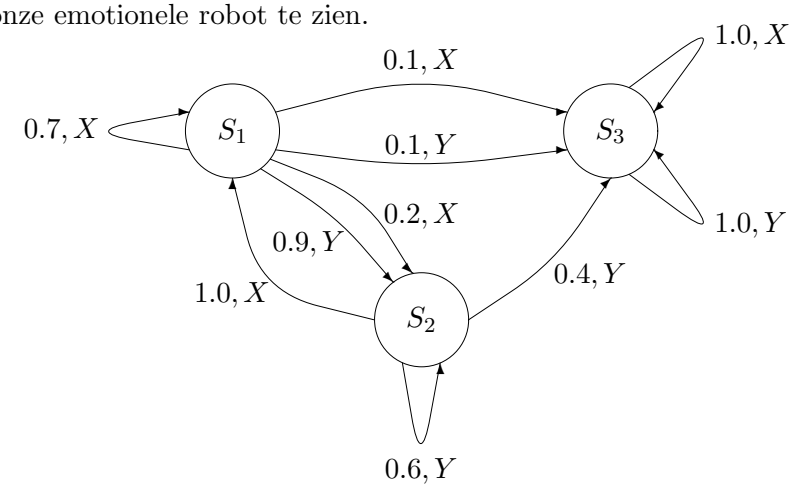
$$A_Y^5 \cdot A_X^2 = \begin{pmatrix} 0.08163 & 0.02332 & 0.8950 \\ 0.05443 & 0.01555 & 0.9300 \\ 0.0 & 0.0 & 1.0 \end{pmatrix}$$

dus is zelfs een gelukkige robot na afloop van een week (en na een eigenlijk opbouwend weekend) met hoge kans mal. Nog erger is het van vrijdag tot vrijdag, dit geeft de overgangskansen

$$A_X^2 \cdot A_Y^5 = \begin{pmatrix} 0.0 & 0.09135 & 0.9086 \\ 0.0 & 0.09718 & 0.9028 \\ 0.0 & 0.0 & 1.0 \end{pmatrix}$$

en alleen maar de robots die op vrijdag middag bedroefd en nog niet mal zijn, krijgen we tot maandag weer opgeknapt.

Een stochastische automaat laat zich analoog met een Markov proces van orde 1 door een state diagram beschrijven, waarbij de labels de input en de kans voor de overgang bij deze input bevatten. In Figuur III.4 is het state diagram voor onze emotionele robot te zien.



Figuur III.4: Stochastische automaat voor een emotionele robot.

Merk op dat in een state diagram voor elke state de som van de kansen op de uitgaande pijltjes voor eenzelfde input gelijk aan 1 moet zijn.

12.3 Markov modellen

We hebben tot nu toe het standpunt ingenomen, dat een rij states volgens de kansverdelingen van een Markov proces voortgebracht wordt. Maar we kunnen de opzet van een Markov proces ook opvatten als een *model* voor een niet verder gespecificeerd mechanisme dat de rij van states voortbrengt. Uit deze perspectief noemt men het stelsel van states en kansverdelingen voor de overgangen tussen de states een *Markov model*. Het idee hier achter is dat een onbekend proces de states produceert, maar dat we veronderstellen dat dit proces zich gedraagt als een Markov proces en de states en overgangskansen dus een model voor het proces zijn.

Om voor een onbekend proces een Markov model te maken, moeten we uit waarnemingen van rijen van states de overgangskansen tussen de states schatten. Hoe dit in zijn werk gaat, bekijken we aan het voorbeeld van de letterfrequenties:

Markov model van **orde 0**:
We hebben alleen maar de kansverdeling van de states nodig, dus de kansverdeling van de letters, en die krijgen we als relatieve frequenties van de letters in een (grote) achtergrond tekst (training tekst).

Markov model van **orde 1**:
We moeten de overgangskansen $a_{ij} := p(q_t = S_j \mid q_{t-1} = S_i)$ bepalen. Maar er geldt voor de voorwaardelijke kans a_{ij} dat

$$p(q_t = S_j \mid q_{t-1} = S_i) = \frac{p(q_t = S_j, q_{t-1} = S_i)}{p(q_{t-1} = S_i)},$$

dus kunnen we de a_{ij} op een training tekst bepalen als quotiënt van de frequentie f_{ij} van letterparen met S_i als eerste letter en de totale frequentie f_i van de letter S_i. Hierbij hoeven we frequenties f_i van de enkele letters niet eens expliciet te bepalen, want er geldt $f_i = \sum_j f_{ij}$ omdat we elk voorkomen van S_i hebben geteld als we alle paren met S_i in de eerste plaats hebben geteld. (Voor de letter op de laatste plaats in de training tekst klopt dit natuurlijk niet, maar deze fout kunnen we verwaarlozen). We krijgen dus de overgangskansen a_{ij} heel makkelijk als

$$a_{ij} = \frac{f_{ij}}{f_i}.$$

Markov model van **orde k** ≥ 2:
In principe passen we hier hetzelfde idee toe als bij een Markov model van orde 1 en berekenen de voorwaardelijke kansen $p(q_t = S_j \mid q_{t-1} = S_{i_k}, \ldots, q_{t-k} = S_{i_1})$ door

$$p(q_t = S_j \mid q_{t-1} = S_{i_k}, \ldots, q_{t-k} = S_{i_1}) = \frac{p(q_t = S_j, q_{t-1} = S_{i_k}, \ldots, q_{t-k} = S_{i_1})}{p(q_{t-1} = S_{i_k}, \ldots, q_{t-k} = S_{i_1})}.$$

De kans in de teller vinden we hierbij als relatieve frequentie van de rij van states $(S_{i_1}, \ldots, S_{i_k}, S_j)$ in alle rijen van $k+1$ states en de kans in de noemer als relatieve frequentie van de rij $(S_{i_1}, \ldots, S_{i_k})$ in alle rijen van k states.

> In de aanpak met de relatieve frequenties bestaat er een klein probleem met de zogeheten *zeldzame gebeurtenissen*. Voor een rij van states met een lage kans kan het gebeuren dat deze rij in het training materiaal helemaal niet voorkomt. Maar in het algemeen is het niet verstandig om aan een overgang in het model de kans 0 toe te kennen, omdat dit betekent dat het model deze overgang nooit zou produceren en aan een rij states waarin deze overgang wel voorkomt de kans 0 geeft. De enige uitzondering zijn *verboden overgangen*, d.w.z. overgangen die uit inhoudelijke redenen inderdaad uitgesloten kunnen worden (bijvoorbeeld in een populatie kippen de overgang van een overleden kip tot een vruchtbaar kip).
>
> Een simpele (maar vaak voldoende) oplossing van het probleem van de zeldzame gebeurtenissen is, de teller voor de frequenties van de rijen niet met 0 maar met 1 te initialiseren, dus te veronderstellen dat elk gebeurtenis wel een keer is gezien (zo iets als éénmaal is geen maal). Maar

er zijn ook ingewikkeldere, theoretisch beter onderbouwde oplossingen voor dit probleem bedacht, dit valt onder het begrip van *smoothing* (gladmaken) van kansverdelingen.

We hebben gezien dat Markov processen en Markov modellen in principe twee zijden van eenzelfde munt zijn: Uit waarnemingen van een onbekend proces maken we een Markov model, en we zeggen dat het Markov model het proces goed beschrijft als de Markov proces die bij het Markov model hoort een rij gebeurtenissen produceert dat goed met de waarnemingen overeen komt.

12.4 Toepassingen van Markov modellen

De twee perspectieven om naar Markov modellen te kijken geven ook de meest belangrijke typen van toepassingen: Simulatie met behulp van Markov processen en classificatie (of toetsing) met behulp van Markov modellen.

Simulatie

Hierbij gebruiken we een Markov model om een rij gebeurtenissen voort te brengen, waarop bijvoorbeeld andere modellen getoetst kunnen worden.

Als we bijvoorbeeld voor de rijen van letters in verschillende talen een Markov model van orde 1 bouwen, kunnen we (onzinnige) teksten produceren, die niettemin typische elementen van de taal laten zien.

Voor de talen Nederlands (NL), Engels (EN), Duits (DU) en Fins (FI) krijgen we zo bijvoorbeeld *teksten* als de volgende (met 160 letters):

NL `EVEFOOE OVORER KET DESTS NDEFT MELL CEN HEN ET MEDE ENIJFEBE HEPGE G IN JEN VOONDEDE HE ESTETETE DE HE DER COROPEETLL NFFTE LENG MHT VOT HET EUDE DERANLODENGEMH`

EN `S COTHENN CHENCTHER BEN THXS INTHABJ IT EUPAUS ISTHANTEN CIOPE WAGESON IN M CONA ATHEDEDED AN JON DERENN T RTH THEPLE UES PTAD TIONTHAT ERO OR FFION TTUNEROCTHE`

DU `RELEMM FT DELLATIT APTZERKO TUN ASER WOPF KPEH RARINTOKEN IG W MT BURER MENGS URHEM ZICHAAT KAHED URIIENSP ENTEN ERT ZUNAUN SIONG D SE VERZUR HUMAN TSER DIE ASC`

FI `VA EN MMA LLLEN LILIOD TOS IHTORON ATUN MISA VUN KA OROLUSAMUJA POKUNITUSIM M DOSTOTA HAITTANEMINTISISON URECD KOMI HTI KUOHOONTOULI T OUUUSKARIS OP SSEHJOITAVU`

Deze teksten laten verschillende elementen zien, die typisch voor de talen zijn, zo als de dubbele `OO` en de `IJ` in het Nederlands of de `TH` in het Engels.

Als men hier in plaats van een Markov model van orde 1 een Markov model van orde 2 toepast, dus de relatieve frequenties van tripels van letters telt, worden de verschillen nog veel duidelijker. Merk op dat er $27^3 = 19683$ verschillende tripels van letters zijn, om hiervoor een redelijke kansverdeling te krijgen, zou men een training tekst van een paar miljoen letters nodig hebben (voor de

$27^2 = 729$ paren van letters zijn training teksten van slechts ongeveer 50000 letters gebruikt). Maar natuurlijk zijn zo grote teksten voor alle soorten van talen beschikbaar, en als een Markov model van orde 2 een tekst als

```
IBUS CENT IPITIA IPSE CUM VIVIVS SE ACETITI DEDENTUR
```

produceert, zouden we er snel achter komen, dat het model op Latijnse teksten getraind is.

Classificatie/Toetsing

We veronderstellen dat we bij een classificatie taak in de patroonherkenning voor elke klasse van patronen een Markov model gebouwd hebben dat de elementen van de klasse goed beschrijft.

Classificatie principe: Voor een gegeven rij waarnemingen wordt voor iedere klasse van patronen berekend met welke kans de waarneming door het Markov model van deze klasse voortgebracht wordt. Het patroon wordt aan de klasse toegewezen, waarvoor deze kans maximaal is.

Dit principe laat zich als volgt onderbouwen:

Bij een Markov proces van orde 0 met N mogelijke uitkomsten $S_1, \ldots, S_N$ kunnen we de kans op de rij $x_1x_2 \ldots x_n$ van uitkomsten eenvoudig berekenen door

$$p(x_1x_2 \ldots x_n) = p(x_1) \cdot p(x_2) \cdot \ldots \cdot p(x_n).$$

We bekijken nu de stochast X van een Markov proces met *echte* kansen $p_i = p(X = S_i)$ en beschrijven deze door een Markov model met (geschatte) kansen $q_i = p'(q_t = S_i)$.

Als n groot is, is het aantal van uitkomsten x_i in de rij ongeveer gelijk aan $n \cdot p_i$. Dan krijgen we

$$p(x_1x_2 \ldots x_n) = \prod_{i=1}^{N} p(X = S_i)^{n \cdot p_i} = \prod_{i=1}^{N} p_i^{n \cdot p_i}$$

voor de juiste kansen en

$$p'(x_1x_2 \ldots x_n) = \prod_{i=1}^{N} p'(q_t = S_i)^{n \cdot p_i} = \prod_{i=1}^{N} q_i^{n \cdot p_i}$$

voor de kansen volgens het model.

Om rijen van verschillende lengtes te kunnen vergelijken moeten we hieruit nog de n-de machtswortel trekken, dit geeft

$$p(x_1x_2 \ldots x_n)^{\frac{1}{n}} = \prod_{i=1}^{N} p_i^{p_i} \quad \text{tegenover} \quad p'(x_1x_2 \ldots x_n)^{\frac{1}{n}} = \prod_{i=1}^{N} q_i^{p_i}.$$

Als we van deze vergelijkingen de logaritme (met basis 2) nemen, krijgen we een verband met een oude bekende uit de laatste les, namelijk de entropie:

$$\begin{aligned} H(X) &= -\sum_{i=1}^{N} p_i \ {}^2\log(p_i) = -\frac{1}{n} \ {}^2\log(p(x_1x_2\ldots x_n)) \\ &\leq -\sum_{i=1}^{N} p_i \ {}^2\log(q_i) = -\frac{1}{n} \ {}^2\log(p'(x_1x_2\ldots x_n)) =: H. \end{aligned}$$

In de limiet $n \to \infty$ geeft dus $H := -\frac{1}{n} \ {}^2\log(p'(x_1x_2\ldots x_n))$ een schatting voor de entropie $H(X)$ van de kansverdeling van de stochast X en deze schatting is beter naarmate H een lagere waarde heeft, want we weten dat het minimum bereikt wordt als Q de juiste kansverdeling van X is.

Wat we net hebben gezien, laat zich op algemene Markov processen veralgemenen, voor de entropie van een stochast X geldt:

$$H(X) = \lim_{n\to\infty} -\frac{1}{n} \ {}^2\log(p(x_1x_2\ldots x_n))$$

waarbij $p(x_1x_2\ldots x_n)$ de juiste kansverdeling voor de stochast X aangeeft. Als we nu dezelfde kans met de kansen uit een Markov model berekenen, wordt deze kans hoger als het Markov model de stochast beter beschrijft, want voor een hogere kans $p'(x_1x_2\ldots x_n)$ is $-\frac{1}{n} \ {}^2\log(p'(x_1x_2\ldots x_n))$ kleiner en ligt dus dichter bij $H(X)$.

Een andere manier om tot dezelfde conclusie te komen berust op de interpretatie van $2^{H(X)}$ als het gemiddelde aantal alternatieven dat men voor de stochast X verwacht:

Uit de vorige les weten we dat een stochast X met entropie $H(X)$ net zo moeilijk is als een uniforme verdeling met $2^{H(X)}$ alternatieven. Maar we weten dat voor $H = -\frac{1}{n} \ {}^2\log(p'(x_1x_2\ldots x_n))$ steeds geldt dat $H \geq H(X)$ en dus ook $2^H \geq 2^{H(X)}$. We kunnen dus zeggen, dat de beschrijving van de stochast X door het Markov model met kansverdeling Q net zo moeilijk is als een uniforme verdeling met 2^H alternatieven, en natuurlijk is degene beschrijving het beste waarvoor 2^H minimaal is.

We passen dit idee nu op korte testteksten toe, waarvoor we de taal willen bepalen. We nemen aan dat we een Markov model van orde 1 hebben met overgangskansen a_{ij} van state S_i naar state S_j en met kans $b_i = \sum_{j=1}^{N} a_{ij}$ voor state S_i. Met zo'n model berekenen we de kans van een rij $x_1x_2\ldots x_n$ van letters door

$$\begin{aligned} &p(x_1x_2\ldots x_n) \\ &\quad = p(q_1 = S_{i_1}) \cdot p(q_2 = S_{i_2} \mid q_1 = S_{i_1}) \cdot \ldots \cdot p(q_n = S_{i_n} \mid q_{n-1} = S_{i_{n-1}}) \\ &\quad = b_{i_1} \cdot \prod_{j=1}^{n-1} a_{i_j i_{j+1}} \end{aligned}$$

waarbij S_{i_j} de state van de letter x_j is.

Voorbeeld: We onderzoeken verschillende stukken tekst in de talen Nederlands (NL), Engels (EN), Duits (DU) en Fins (FI) met Markov modellen voor deze talen en berekenen voor elke combinatie van tekst en Markov model de waarde

$$2^H \text{ voor } H = -\frac{1}{n}\ {}^2\log(p(x_1x_2\ldots x_n))$$

waarbij de kansen zo als net aangegeven berekend worden.

De testteksten zijn:

T_1 : `SINTERKLAAS KOMT NAAR ONS HUIS`

T_2 : `SANTA CLAUS COMES TO OUR HOUSE`

T_3 : `NIKOLAUS KOMMT IN UNSER HAUS`

T_4 : `HANNU MANNINEN`

Als resultaat krijgen we de volgende tabel met de waarden van 2^H:

	NL	EN	DU	FI
T_1	14.1	28.3	16.2	19.0
T_2	18.2	12.4	28.9	18.0
T_3	14.4	23.5	9.8	16.5
T_4	19.2	25.0	16.8	14.0

Het is duidelijk dat we in elk geval de juiste taal kunnen achterhalen. Hoe typisch de testteksten voor de enkele talen zijn, kunnen we zien als we de boven gevonden waarden met de waarden op de teksten vergelijken waarop de Markov modellen getraind zijn, dus met de entropieën van de Markov modellen zelfs. De waarden van $2^{H(X)}$ voor de verschillende talen zijn:

NL: 9.2 EN: 9.6 DU: 9.3 FI: 9.7.

> De classificatie met behulp van Markov modellen voor letter strings in de verschillende talen is de manier hoe in tekstverwerkingsprogramma's als WORD (OFFICE) automatisch de spellchecker naar een andere taal omgeschakeld wordt, als er bijvoorbeeld in een Nederlandstalige tekst een citaat in het Engels ingebouwd wordt.

12.5 Markov modellen met verborgen states

Tot nu toe hebben we steeds naar systemen gekeken, waarvoor we de states direct konden waarnemen. We hebben daarom ook geen onderscheiding gemaakt tussen states, uitkomsten en waarnemingen. We krijgen echter een grotere vrijheid in de Markov modellen, als we de states los van de gebeurtenissen en waarnemingen bekijken. Het idee is, dat de states de mogelijke uitkomsten wel veroorzaken, maar dat verschillende states dezelfde uitkomst kunnen produceren en dat niet (noodzakelijk) bekend is, welke state een bepaalde uitkomst heeft veroorzaakt. Om deze reden noemen we de states ook *verborgen* en een

Markov model met verborgen states heet een *Hidden Markov model*, of in het kort een *HMM*.

We geven twee opzetten die het idee van de Hidden Markov modellen illustreren:

- **Het munt model**

  Achter een gordijn zit iemand die met een aantal mogelijk geladen (dus niet noodzakelijk eerlijke) munten een muntworp experiment uitvoert, maar alleen maar de rij uitkomsten (kop/munt) aan de waarnemer doorgeeft. De keuze van de munten voor de enkele worpen volgt een stochastisch proces die door overgangskansen tussen de munten bepaald wordt.

- **Het vaas model**

  Er zijn N vazen met telkens ballen van M verschillende kleuren, waarbij de aantallen van ballen met een zekere kleur per vaas mogen verschillen en ook het totale aantal ballen per vaas niet hetzelfde hoeft te zijn. Iemand trekt (met terugleggen) een bal uit een van de vazen en geeft de kleur van de bal aan de waarnemer door. Vervolgens wordt volgens een toevalskeuze, die afhankelijk van de laatst gekozen vaas is, een nieuwe vaas gekozen.

De algemene ingrediënten van een HMM (van orde 1) zijn als volgt:

(1) Mogelijke uitkomsten $x_1, \ldots, x_M$. De waargenomen uitkomst op tijdstip t word met o_t aangegeven (de letter o staat voor het Engelse *observation*).

(2) Een aantal states $S_1, \ldots, S_N$, waarbij de state op tijdstip t met q_t aangegeven wordt.

(3) De overgangskansen $a_{ij} := p(q_t = S_j \mid q_{t-1} = S_i)$ voor de overgang van state S_i naar state S_j.

(3) Voor elke state S_i een kansverdeling b_i voor de *emissiekansen*, d.w.z. $b_i(x_k) = p(o_t = x_k \mid q_t = S_i)$ is de kans dat in state S_i de uitkomst x_k geproduceerd wordt. Er wordt veronderstelt dat deze kansen onafhankelijk van het tijdstip t zijn.

(4) Een beginverdeling π die de kansen $\pi(i) := p(q_0 = S_i)$ aangeeft dat het systeem op tijdstip $t = 0$ in state S_i is.

> Ook een gewoon Markov model laat zich (op een iets kunstmatige manier) als HMM opvatten: Hiervoor worden de states S_i identiek met de uitkomsten x_i gekozen en de emissiekansen b_i worden gedefinieerd door
>
> $$b_i(x_i) = 1 \text{ en } b_i(x_k) = 0 \text{ voor } k \neq i.$$

Voorbeeld van een HMM

We bekijken een munt model met drie munten als states, waarvan de eerste eerlijk is, dus kansen $\frac{1}{2}$ voor *kop* en *munt* heeft, de tweede oneerlijk met kans $\frac{3}{4}$ op

kop en de derde oneerlijk met kans $\frac{1}{4}$ op *kop*. Als we `K` voor de uitkomst *kop* en `M` voor de uitkomst *munt* schrijven, hebben we de emissiekansen $b_1(\texttt{K}) = b_1(\texttt{M}) = \frac{1}{2}$, $b_2(\texttt{K}) = b_3(\texttt{M}) = \frac{3}{4}$, $b_3(\texttt{K}) = b_2(\texttt{M}) = \frac{1}{4}$, die door de volgend tabel weergegeven worden:

	$b_i(\texttt{K})$	$b_i(\texttt{M})$
S_1	0.5	0.5
S_2	0.75	0.25
S_3	0.25	0.75

We veronderstellen verder dat de beginverdeling uniform is, d.w.z. de kans dat het systeem in het begin in state S_i is, is voor elke state $\pi(i) = \frac{1}{3}$.

Stel we nemen de rij $O = \texttt{KMKMK}$ waar.

In een eerste opzet nemen we aan dat alle overgangskansen hetzelfde zijn, dus alle $a_{ij} = \frac{1}{3}$.

Omdat de hoogste kans op de uitkomst `K` in state S_2 zit, de hoogste kans op de uitkomst `M` in S_3 en de overgangskansen alle hetzelfde zijn, kunnen we makkelijk zien dat de rij $q = S_2S_3S_2S_3S_2$ de rij van states is, waarvoor de kans op de waarneming O maximaal is. In dit geval is deze kans namelijk $p(O, q) = (\frac{1}{3})^5 \cdot (\frac{3}{4})^5 = (\frac{1}{4})^5 \approx 9.77 \cdot 10^{-4}$.

In tegenstelling hiermee is de kans dat deze waarneming door de rij $q' = S_1S_1S_1S_1S_1$ voortgebracht is, slechts $p(O, q') = (\frac{1}{3})^5 \cdot (\frac{1}{2})^5 = (\frac{1}{6})^5 \approx 1.29 \cdot 10^{-4}$. Deze kans is om een factor $(\frac{3}{2})^5 \approx 7.6$ kleiner dan voor de eerdere rij q van states.

Het probleem wordt iets ingewikkelder als de overgangskansen niet meer alle hetzelfde zijn. Stel we hebben de volgende matrix $A = (a_{ij})$ van overgangskansen a_{ij} tussen de states:

$$A = (a_{ij}) := \begin{pmatrix} 0.9 & 0.05 & 0.05 \\ 0.45 & 0.1 & 0.45 \\ 0.45 & 0.45 & 0.1 \end{pmatrix}$$

dan is de kans $p(O, q \mid A)$ (we geven hier voor de duidelijkheid de matrix van overgangskansen mee aan) voor dezelfde rijen q en q' van states als boven gegeven door

$$p(O, q \mid A) = \frac{1}{3} \cdot 0.45^4 \cdot (\frac{3}{4})^5 \approx 3.24 \cdot 10^{-3},$$
$$p(O, q' \mid A) = \frac{1}{3} \cdot 0.9^4 \cdot (\frac{1}{2})^5 \approx 6.83 \cdot 10^{-3},$$

dus is deze keer $p(O, q' \mid A)$ om een factor $2^4(\frac{2}{3})^5 \approx 2.1$ *groter* dan $p(O, q \mid A)$.

We zien dus dat in het tweede geval de hypothese dat het systeem door de rij q' van states gelopen is, een hogere kans voor de waarneming geeft dan de rij q van states.

Het is nu natuurlijk een voor de hand liggende vraag, of er een verdere rij q'' van states is, die een nog hogere kans voor de rij O van waarnemingen oplevert.

Voor korte rijen kunnen we dit met brute kracht nog wel achterhalen (voor het voorbeeld met 5 waarnemingen en 3 states zijn er $3^5 = 243$ mogelijkheden voor de rij q van states), maar voor langere rijen is dit ondoenlijk.

In het speciaal geval van het voorbeeld is de rij q' inderdaad optimaal, omdat de overgangskans $a_{11} = 0.9$ minstens twee keer groter is dan alle andere overgangskansen en de emissiekansen $b_1(\mathtt{K}) = b_1(\mathtt{M}) = \frac{1}{2}$ zijn. Maar zo'n soort redenering zal in de praktijk natuurlijk nooit werken, omdat de modellen veel ingewikkelder en onoverzichtelijker zijn.

We zitten dus met de vraag hoe we bij een rij waarnemingen de rij states vinden, die de hoogste kans aan de waarnemingen geeft. Dit is één van drie fundamentele problemen in het kader van Hidden Markov modellen die we in de volgende les gaan bespreken.

Belangrijke begrippen in deze les

- Markov processen
- overgangsmatrix
- state diagram
- stochastische automaat
- Markov model
- Hidden Markov model (HMM)

Opgaven

94. In een communicatie systeem worden bits als 0 of 1 over een aantal stappen doorgegeven, waarbij in iedere stap een bit met kans 0.8 correct blijft.

    (i) Beschrijf het communicatie systeem als een Markov proces en geef het state diagram van het proces aan.

    (ii) Bepaal de kans dat een bit met de waarde 0 na vier stappen als 0 ontvangen wordt.

95. De oogst van appels in Tasmanië wordt als *geweldig*, *middelmatig* of *slecht* geclassificeerd. Na een geweldig jaar zijn de kansen voor het volgende jaar 0.5, 0.3, 0.2 voor een geweldige, middelmatige of slechte oogst. Na een middelmatig jaar zijn de kansen voor het volgende jaar 0.2, 0.5, 0.3 en na een slecht jaar zijn de kansen 0.2, 0.2, 0.6 voor een geweldige, middelmatige of slechte oogst.

    (i) Beschrijf de ontwikkeling van de appel oogst door een Markov proces en geef het state diagram van het proces aan.

    (ii) Stel de kansen om met een geweldig, middelmatig of slecht jaar te beginnen zijn 0.2, 0.5 en 0.3. Wat zijn de kansverdelingen voor de kwaliteit van de oogst na 1 jaar, 3 jaren en 5 jaren?

(iii) Kan je de kansverdeling voor de kwaliteit van de oogst bepalen, die op lange termijn bereikt wordt?

96. Een Markov proces heet *irreducibel* als elke state in eindig veel stappen vanuit elke andere state bereikbaar is. Laat zien dat de Markov processen met overgangsmatrices

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0.5 & 0 & 0.5 \\ 1 & 0 & 0 \end{pmatrix} \quad \text{en} \quad B = \begin{pmatrix} 0 & 0 & 0.5 & 0.5 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

irreducibel zijn.

97. We bekijken de emotionele robot uit sectie 12.2 en bepalen de kansverdeling voor zijn toestand na twee inputs.

(i) Veronderstel dat de robot in het begin gelukkig is en bereken de kansverderdeling voor elk van de vier mogelijke inputs XX, XY, YX en YY.

(ii) Bereken de kansverdelingen voor de verschillende inputs ook voor de gevallen dat de robot in het begin bedroefd of mal was.

98. De states S_1, S_2, S_3 van een Hidden Markov model zijn (net als in het voorbeeld) drie munten die de emissiekansen $\frac{1}{2}, \frac{3}{4}, \frac{1}{4}$ op kop (`K`) en de emissiekansen $\frac{1}{2}, \frac{1}{4}, \frac{3}{4}$ op munt (`M`) hebben. De beginverdeling van de states is uniform, dus $\pi(1) = \pi(2) = \pi(3) = \frac{1}{3}$. We bekijken de drie rijen waarnemingen $O_1 =$ `KKKK`, $O_2 =$ `KKKM`, $O_3 =$ `KKMM`.

(i) Veronderstel dat alle overgangskansen hetzelfde zijn, dus gelijk aan $\frac{1}{3}$. Bepaal de rijen q_1, q_2, q_3 van states, waarvoor de kans dat zij de waarnemingen O_1, O_2, O_3 geproduceerd hebben maximaal is. Bereken voor de gevonden rijen van states de kansen $p(O_1, q_1)$, $p(O_2, q_2)$, $p(O_3, q_3)$.

(ii) Vergelijk de kansen uit (i) met de kansen $p(O_i, q)$ die men krijgt, als men aanneemt dat altijd de eerlijke munt geworpen wordt, dus als $q = S_1S_1S_1S_1$ is.

(iii) Veronderstel nu dat de overgangskansen niet uniform zijn, maar gegeven door de matrix

$$A = (a_{ij}) := \begin{pmatrix} 0.6 & 0.2 & 0.2 \\ 0.4 & 0.2 & 0.4 \\ 0.4 & 0.4 & 0.2 \end{pmatrix}.$$

Bereken de kansen $p(O_i, q_i \mid A)$ voor de rijen van states uit deel (i) en de kansen $p(O_i, q \mid A)$ voor de rij q van states uit deel (ii) met betrekking tot deze overgangskansen.

(iv) Probeer in deel (iii) de rijen q_1', q_2', q_3' van states te vinden, zo dat $p(O_i, q_i' \mid A)$ maximaal wordt.

Les 13 Hidden Markov modellen

In deze les zullen we nader op Hidden Markov modellen ingaan, in het bijzonder op de technieken en algoritmen die bij het omgaan met dit soort modellen belangrijk zijn. Om de notaties helder te hebben, spreken we nu af dat we een Hidden Markov model als volgt beschrijven:

Een Hidden Markov model (vanaf nu afgekort als HMM) λ is gegeven door $\lambda = \lambda(\mathcal{S}, \mathcal{X}, A, B, \pi)$, waarbij de parameters de volgende betekenis hebben:

- $\mathcal{S} = \{S_1, \ldots, S_N\}$ is een verzameling van states;
- $\mathcal{X} = \{x_1, \ldots, x_M\}$ is een verzameling van uitkomsten, die door de states geproduceerd worden;
- $A = (a_{ij})$ is de matrix van overgangskansen tussen de states, d.w.z. $a_{ij} = p(q_t = S_j \mid q_{t-1} = S_i)$ is de kans voor de overgang van state S_i naar state S_j (onafhankelijk van het tijdstip t);
- $B = b_i(k)$ is de matrix van emissiekansen voor de gebeurtenissen vanuit de states, d.w.z. $b_i(k) = p(o_t = x_k \mid q_t = S_i)$ is de kans dat de state S_i de uitkomst x_k produceert (onafhankelijk van t);
- $\pi = (\pi(1), \ldots, \pi(N))$ is de beginverdeling van de states.

Vaak behoren de states en de gebeurtenissen tot de algemene opzet van een probleem, in dit geval staan alleen maar de verschillende kansverdelingen ter discussie. In zo'n geval wordt een HMM iets korter door $\lambda = \lambda(A, B, \pi)$ beschreven.

Er zijn in feite drie fundamentele vragen, waarmee we ons moeten bemoeien:

(1) Gegeven een rij $O = o_1 o_2 \ldots o_T$ van waarnemingen en een HMM $\lambda = \lambda(A, B, \pi)$, hoe vinden we de kans $p(O \mid \lambda)$ op deze waarnemingen, gegeven het model λ? Deze kans kan men ook interpreteren als maat, hoe goed het model bij de waarnemingen past.

(2) Gegeven een rij $O = o_1 o_2 \ldots o_T$ van waarnemingen en een HMM $\lambda = \lambda(A, B, \pi)$, hoe vinden we de rij $q = q_1 q_2 \ldots q_T$ van states die de rij waarnemingen het beste kan verklaren?

(3) Hoe kunnen we de parameters van het HMM $\lambda = (A, B, \pi)$ zo aanpassen dat $p(O \mid \lambda)$ voor een (vaste) rij O van waarnemingen maximaal wordt?

De eerste vraag gaat over het evalueren van een gegeven model op een rij waarnemingen, de tweede over het onthullen van de verborgen states en de derde over het vinden van de parameters van een HMM, zo dat het model goed bij een gegeven rij waarnemingen past. Het laatste noemt men ook het *training* van een HMM. We zullen deze vragen nu apart bekijken.

13.1 Evalueren met behulp van een HMM

Stel we hebben een rij waarnemingen $O = o_1 o_2 \dots o_T$ en een HMM $\lambda = \lambda(A, B, \pi)$ en we willen de kans $p(O \mid \lambda)$ op de rij waarnemingen, gegeven het model, berekenen. Een typische situatie waar men dit probleem tegen komt is de classificatie van de waarneming O. Stel dat verschillende klassen $C_1, \dots, C_r$ door verschillende HMM's $\lambda_1, \dots, \lambda_r$ gekarakteriseerd zijn, dan is het een voor de hand liggende idee de waarneming O aan degene klasse C_k toe te wijzen, waarvoor $p(O \mid \lambda_k)$ maximaal is. Deze aanpak noemt men ook de *maximum likelihood* methode.

Om de kans $p(O \mid \lambda)$ te berekenen moeten we in principe voor elke rij $q = q_1 q_2 \dots q_T \in \mathcal{S}^T$ van states de kans $p(O, q \mid \lambda)$ berekenen en deze kansen voor alle mogelijke rijen q van states bij elkaar optellen. Volgens de definitie van de voorwaardelijke kans geldt

$$p(O, q \mid \lambda) = p(O \mid q, \lambda) \cdot p(q \mid \lambda)$$

en dus

$$p(O \mid \lambda) = \sum_{q \in \mathcal{S}^T} p(O, q \mid \lambda) = \sum_{q \in \mathcal{S}^T} p(O \mid q, \lambda)\, p(q \mid \lambda).$$

Met behulp van de laatste uitdrukking kunnen we de kans $p(O \mid \lambda)$ inderdaad uitrekenen: Aan de ene kant is $p(q \mid \lambda)$ juist het product van de kansen voor de overgangen tussen de states in de rij $q = q_1 q_2 \dots q_T$, dus

$$p(q \mid \lambda) = \pi(q_1) \cdot \prod_{t=1}^{T-1} a_{q_t q_{t+1}}.$$

Aan de andere kant is voor een gegeven rij van states de kans $p(O \mid q, \lambda)$ het product van de emissiekansen van de enkele states, dus

$$p(O \mid q, \lambda) = \prod_{t=1}^{T} b_{q_t}(o_t).$$

Bij elkaar genomen krijgen we zo:

$$\begin{aligned} p(O \mid \lambda) &= \sum_{q = q_1 \dots q_T} \pi(q_1) b_{q_1}(o_1) \prod_{t=1}^{T-1} a_{q_t q_{t+1}} b_{q_{t+1}}(o_{t+1}) \\ &= \sum_{q = q_1 \dots q_T} \pi(q_1) b_{q_1}(o_1) a_{q_1 q_2} b_{q_2}(o_2) \dots a_{q_{T-1} q_T} b_{q_T}(o_T). \end{aligned}$$

Het probleem hierbij is, dat we voor een rij van lengte T over N^T mogelijke rijen van states moeten lopen, en dit is al voor kleine waarden van T (bijvoorbeeld $T = 100$) ondoenlijk.

Gelukkig kunnen we het vermijden over alle mogelijke rijen van states te lopen. Bij de brute kracht methode zouden we erg veel dingen herhaaldelijk uitrekenen, namelijk de beginstukken van de rijen waarvoor de eerste t states hetzelfde zijn. Het idee is, de kansen voor de beginstukken stapsgewijs te

berekenen en deze te recyclen. Als we namelijk de kans voor het beginstuk $o_1 o_2 \dots o_t$ al kennen, zijn er maar N mogelijkheden voor de state waarin het systeem op tijdstip t zit, en voor de voortzetting naar o_{t+1} hoeven we alleen maar de overgangen van deze N mogelijkheden naar de N mogelijke states op tijdstip $t+1$ te berekenen. Zo krijgen we slechts $T \cdot N^2$ waarden, die we moeten berekenen. De procedure die we zo net hebben geschetst is zo belangrijk dat ze een eigen naam heeft (ook al is die niet erg karakteristiek), ze heet *forward algoritme*.

Forward algoritme

We willen voor $O = o_1 o_2 \dots o_T$ de kans $p(O \mid \lambda)$ berekenen. Hiervoor definiëren we de *vooruitkans*

$$\alpha_t(i) := p(o_1 o_2 \dots o_t, q_t = S_i \mid \lambda),$$

die de kans aangeeft dat het systeem op tijdstip t in state S_i is en tot dit tijdstip de waarnemingen $o_1, \dots, o_t$ heeft geproduceerd.

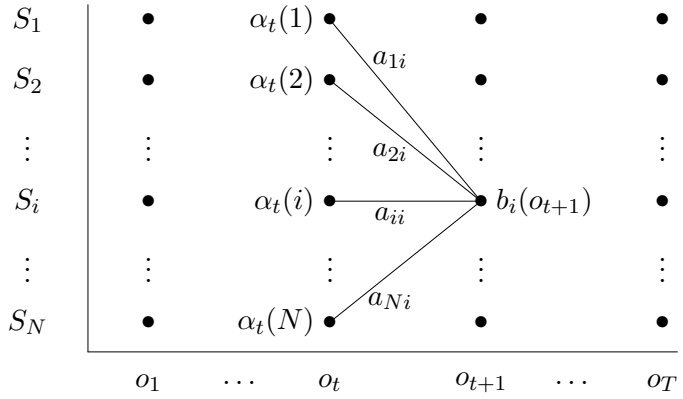
Voor $t = 1$ laten zich de vooruitkansen $\alpha_1(i)$ heel eenvoudig berekenen, er geldt

$$\alpha_1(i) = \pi(i) b_i(o_1).$$

Als we nu van tijdstip t naar tijdstip $t+1$ willen, moeten we over alle N states waarin het systeem op tijdstip t kan zijn lopen en de kans op de overgang naar de verschillende states op tijdstip $t+1$ en de emissie van waarneming o_{t+1} berekenen. Dit geeft de recursie formule:

$$\alpha_{t+1}(i) = \left(\sum_{k=1}^{N} \alpha_t(k) a_{ki} \right) b_i(o_{t+1}).$$

In Figuur III.5 is de berekening van $\alpha_{t+1}(i)$ in een schema aangegeven: De kansen $\alpha_t(1), \dots, \alpha_t(N)$ van de voorafgaande stap worden met de overgangskansen $a_{1i}, \dots, a_{Ni}$ en de emissiekans $b_i(o_{t+1})$ gecombineerd tot de kans $\alpha_{t+1}(i)$.



Figuur III.5: Berekening van $\alpha_{t+1}(i)$ in het *forward algoritme*.

Als we de vooruitkansen $\alpha_t(i)$ voor $t = 1, 2, \dots, T$ berekenen, hoeven we in de laatste stap alleen maar nog de kansen voor de N verschillende states op

tijdstip $t = T$ op te tellen want omdat het systeem in een van de states moet zijn, geeft dit juist de kans op de volledige rij waarnemingen aan. Op deze manier krijgen we

$$p(o_1 o_2 \dots o_T \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i).$$

Backward algoritme

Het zou geen verrassing zijn dat er behalve van een forward algoritme ook een *backward algoritme* bestaat, waarbij de kansen op een deel van de waarnemingen van het einde af berekend worden. Men definieert de *achteruitkans* $\beta_t(i)$ als de voorwaardelijke kans

$$\beta_t(i) := p(o_{t+1} \dots o_T \mid q_t = i, \lambda)$$

op de laatste $T-t$ waarnemingen $o_{t+1}, \dots, o_T$, gegeven het feit dat het systeem op tijdstip t in state S_i was.

In dit geval heeft men de initialisering $\beta_T(i) = 1$ want we veronderstellen dat het systeem op tijdstip T in state S_i is.

Om van het tijdstip $t+1$ naar t terug te komen, moeten we over alle states lopen waarin het systeem op tijdstip $t+1$ kan zijn en de overgangen en de emissie van de waarneming o_{t+1} vanuit deze states berekenen. Dit geeft de recursie

$$\beta_t(i) = \sum_{k=1}^{N} a_{ik} b_k(o_{t+1}) \beta_{t+1}(k).$$

Door deze recursie voor $t = T-1, \dots, 2, 1$ te doorlopen, krijgen we uiteindelijk de kans $p(O \mid \lambda)$ door

$$p(O \mid \lambda) = \sum_{i=1}^{N} \pi(i) b_i(o_1) \beta_1(i).$$

We zullen de vooruitkansen $\alpha_t(i)$ en de achteruitkansen $\beta_t(i)$ later in deze les nog eens tegenkomen. Door een slimme combinatie van de $\alpha_t(i)$ en $\beta_t(i)$ laten zich namelijk de parameters van een HMM zo verbeteren dat het systeem een hogere kans voor een gegeven rij waarnemingen oplevert. Op deze manier wordt het HMM beter aan de waarnemingen aangepast, dus getraind.

> De combinatie van vooruit- en achteruitkansen speelt ook bij problemen een rol, waar snel een kandidaat voor een rij states met hoge kans gevonden moet worden. Het idee is, tegelijkertijd aan het begin en aan het eind te beginnen, tot dat $\alpha_t(i)$ en $\beta_t(i)$ in het midden op elkaar stoten. Daarbij worden alleen maar de meestbelovende trajecten meegenomen, d.w.z. de states op tijdstip t die de hoogste vooruit- en achteruitkansen hebben. Deze manier om de *zoekruimte* snel tot de interessante states te beperken staat bekend onder de naam *beam-search*.

13.2 States onthullen

Vaak is het niet genoeg de kans voor een rij waarnemingen, gegeven een HMM, te bepalen, men wil ook een rij states bepalen die bij de waarnemingen past. Maar omdat er verschillende rijen states zijn, die een rij waarnemingen kunnen produceren, moet men hier een criterium hebben, welke states het beste passen. Voor dat we erover na kunnen denken hoe we een optimale rij states kunnen vinden, moeten we dus eerst definiëren, wat we met de *optimale rij states* bij een rij waarnemingen überhaupt bedoelen,

Helaas is er geen *juiste* manier, om een optimaliteitscriterium te definiëren, en afhankelijk van het probleem worden ook verschillende criteria gehanteerd.

Een mogelijkheid is bijvoorbeeld, op elke tijdstip t de state $q_t = S_i$ te kiezen die op dit tijdstip optimaal is. Dat wil zeggen we kiezen q_t zo dat $p(O, q_t = S_i \mid \lambda)$ maximaal wordt. Merk op dat we dit met behulp van de vooruit- en achteruitkansen keurig kunnen formuleren, er geldt namelijk dat

$$p(O, q_t = S_i \mid \lambda) = \alpha_t(i)\beta_t(i)$$

en we hoeven dus voor q_t alleen maar de state S_i te kiezen waarvoor $\alpha_t(i)\beta_t(i)$ maximaal wordt.

Soms willen (of kunnen) we voor de state q_t op tijdstip t alleen maar de waarnemingen $o_1 \dots o_t$ tot op dit tijdstip gebruiken, bijvoorbeeld in een *real-time* systeem. In dit geval zouden we de state $q_t = S_i$ zo kunnen kiezen, dat $p(o_1 o_2 \dots o_t, q_t = S_i \mid \lambda)$ maximaal wordt. Maar dit betekent, dat we voor q_t de state S_i kiezen, waarvoor $\alpha_t(i)$ maximaal is, want dit is precies de definitie van de vooruitkans.

Een probleem bij de genoemde criteria is, dat de overgangen tussen de states enigszins buiten beschouwing blijven, en we zo misschien zelfs een rij van states krijgen die een *verboden* overgang bevat, dus een overgang met kans 0.

Het meest gebruikte criterium dat dit probleem voorkomt is, de optimale rij q_{opt} van states te definiëren als de rij waarvoor de gemeenschappelijke kans over de hele rij states en waarnemingen maximaal is.

Criterium: We noemen een rij $q_{opt} = q_1 q_2 \dots q_T \in \mathcal{S}^T$ van states *optimaal* voor de waarneming $O = o_1 o_2 \dots o_T$ als

$$p(O, q_{opt} \mid \lambda) \geq p(O, q \mid \lambda) \text{ voor alle } q \in \mathcal{S}^T.$$

We staan nu weer voor het probleem dat we in principe de kans $p(O, q \mid \lambda)$ voor alle rijen q van states moeten berekenen. Anders als bij het berekenen van de kans voor de waarneming mogen we nu niet alle mogelijkheden om tot een tussenpunt te komen bij elkaar optellen, dus helpen de vooruitkansen $\alpha_t(i)$ hier niet verder.

Maar een kleine variatie van het forward algoritme geeft ook hier een oplossing, waarbij we niet alle N^T mogelijke rijen moeten bekijken. Het idee wat hier achter zit komt uit het *dynamische programmeren* en is een bijna vanzelfsprekende opmerking, maar is wel zo fundamenteel, dat het de naam *Bellman's principe* draagt.

Bellman's principe

We bekijken een iets algemenere situatie die uit het dynamische programmeren ontleend is: Stel we hebben een rooster met punten (i, j) voor $0 \leq i \leq N$, $0 \leq j \leq M$, en we zijn op zoek naar en pad van $(0, 0)$ naar (N, M). Met elke overgang van een punt naar een andere zijn kosten verbonden, die we als *afstanden* tussen de punten zien, daarbij noteren we de kosten voor de overgang van (i', j') naar (i, j) met $d((i', j'), (i, j))$. Sommige van de kosten kunnen oneindig zijn, om uit te drukken dat deze overgang onmogelijk is.

Voor elk punt (i, j) noemt men de punten (i', j') waarvoor de overgang van (i', j') naar (i, j) mogelijk is (d.w.z. eindige kosten heeft) de *mogelijke voorgangers* een het stelsel van mogelijke voorgangers noemt men de *lokale beperkingen*. In sommige toepassingen kan men bijvoorbeeld alleen maar van $(i - 1, j - 1)$, $(i - 1, j)$ of $(i, j - 1)$ naar (i, j) komen, in andere gevallen zijn alle punten $(i - 1, j')$ mogelijke voorgangers van (i, j). Dit is bijvoorbeeld het geval als de eerste coördinaat tijdstippen en de tweede states aangeeft en we veronderstellen dat we van elke state naar elke andere state kunnen komen.

Het optimale pad van $(0, 0)$ naar (N, M) is natuurlijk het pad waarvoor de som van de kosten van de overgangen minimaal is. Bellman's principe zegt nu het volgende:

Bellman's principe: *Als het optimale pad van* $(0, 0)$ *naar* (N, M) *door het punt* (i, j) *loopt, dan is ook het deelpad van* $(0, 0)$ *tot* (i, j) *een optimaal pad tussen deze twee punten, net als het deelpad van* (i, j) *naar* (N, M) *een optimaal pad tussen deze twee punten is.*

Hier zit alleen maar de vanzelfsprekende opmerking achter dat we de kosten voor het pad van $(0, 0)$ via (i, j) naar (N, M) nog kunnen reduceren, als we de kosten voor een van de deelpaden tussen $(0, 0)$ en (i, j) of tussen (i, j) en (N, M) kunnen reduceren.

Maar als gevolg van Bellman's principe krijgen we een efficiënte manier om het optimale pad te vinden. We moeten (afhankelijk van de lokale beperkingen) stapsgewijs de optimale paden voor de punten (i, j) bepalen, door voor elke mogelijke voorganger (i', j') van (i, j) de kosten voor het optimale pad naar (i', j') bij de kosten voor de overgang van (i', j') naar (i, j) op te tellen en het minimum van deze kosten te kiezen.

Viterbi algoritme

Als we Bellman's principe op het probleem van de optimale rij van states van een HMM toepassen, krijgen we het *Viterbi algoritme*. Bellman's principe zegt in dit geval dat voor de optimale rij $q = q_1 q_2 \dots q_T$ van states voor de waarneming $O = o_1 o_2 \dots o_T$ ook de deelrijen tot en vanaf tijdstip t optimaal zijn, dus $p(o_1 \dots o_t, q_1 \dots q_t \mid \lambda)$ is maximaal en $p(o_t \dots o_T, q_t \dots q_T \mid \lambda)$ is maximaal.

In de opzet van het dynamische programmeren hebben we als roosterpunten de paren (t, i) die aangeven dat $q_t = S_i$ is. Hierbij beginnen we met het (formele) punt $(0, 0)$ en eindigen in een punt (T, i), waarbij we geen beperking op i opleggen. De mogelijke voorgangers van (t, i) zijn $(t - 1, k)$ voor alle

$1 \leq k \leq N$. In plaats van kosten praten we nu over kansen, en natuurlijk willen we voor de kansen niet het minimum maar het maximum vinden. De kans die bij de overgang van $(t-1,k)$ naar (t,i) hoort, is de overgangskans a_{ki} van state S_k naar state S_i en de kans $b_i(o_t)$ om in state S_i op tijdstip t de waarneming o_t te produceren. De totale kans voor de overgang $(t-1,k) \to (t,i)$ is dus

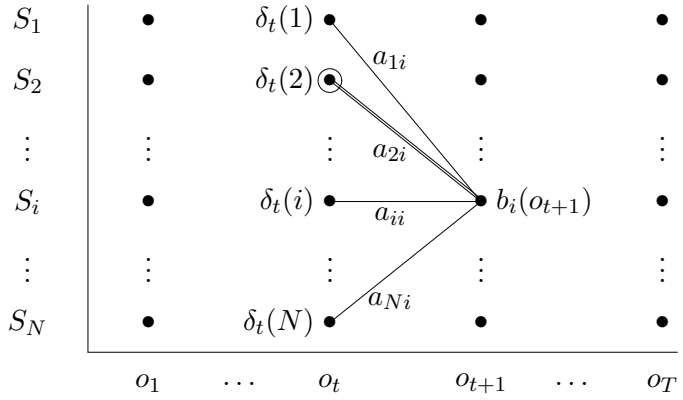
$$p((t-1,k) \to (t,i)) = a_{ki} \cdot b_i(o_t).$$

We definiëren nu $\delta_t(i)$ als de kans van de optimale rij van states voor de deelwaarneming $o_1 o_2 \ldots o_t$, die op tijdstip t in state S_i is.

We krijgen zo de recursie

$$\delta_1(i) = \pi(i) b_i(o_1) \quad \text{en} \quad \delta_{t+1}(i) = \left(\max_{1 \leq k \leq N} \delta_t(k) a_{ki} \right) b_i(o_{t+1})$$

die sterk op de recursie bij het forward algoritme lijkt. Het enige verschil is, dat in plaats van de som over de alle voorgangers nu het maximum over de voorgangers genomen wordt. Maar het schema van het Viterbi algoritme is zo als in Figuur III.6 te zien precies hetzelfde als bij het forward algoritme. Aanvullend moeten we wel bij elke punt (t,i) nog opslaan, vanuit welke voorganger $(t-1,k)$ het maximum bereikt werd, om uiteindelijk het optimale pad terug te kunnen vinden. Dit wordt meestal door een geschakelde lijst geïmplementeerd, in Figuur III.6 is als voorbeeld de overgang $(t,2) \to (t+1,i)$ benadrukt.



Figuur III.6: Berekening van $\delta_{t+1}(i)$ in het Viterbi algoritme.

Om meer efficiëntie bij het evalueren van een rij waarnemingen met verschillende HMM's te bereiken, wordt soms de evaluatie van de kans met behulp van vooruit- of achteruitkansen vervangen door de zogeheten *Viterbi benadering*. Hierbij wordt in plaats van de som over de kansen voor *alle* paden alleen maar de kans voor het beste pad bepaald (en dit natuurlijk met behulp van het Viterbi algoritme). Het idee hierachter is dat bij het evalueren uiteindelijk toch maar heel weinig paden een substantiële bijdrage aan de totale kans leveren en dat de som over de kansen voor hetgeen HMM maximaal wordt waarvoor het optimale pad de hoogste kans heeft.

Er valt nog iets over de implementatie van het Viterbi algoritme (en andere algoritmen in het kader van probabilistische modellen) op te merken:

Omdat er steeds kansen met elkaar vermenigvuldigd worden en deze soms zelf al redelijk klein zijn, worden de waarden van de $\delta_t(i)$ snel erg klein en dalen al gauw onder de rekennauwkeurigheid van een computer. Voor dit probleem bestaat er een heel simpele oplossing: Men rekent met de logaritmen van de kansen.

Merk op: Omdat de logaritme een monotone functie is, is $\delta_t(i)$ maximaal voor de i waarvoor $\tilde{\delta}_t(i) := -\log(\delta_t(i))$ minimaal is.

Als we het Viterbi algoritme op de logaritmen $\tilde{\delta}_t(i) = -\log(\delta_t(i))$ transformeren, krijgen we:

$$\begin{aligned}\tilde{\delta}_1(i) &= -\log(\pi(i)) - \log(b_i(o_1));\\ \tilde{\delta}_{t+1}(i) &= \min_{1\le k\le N}\Big(\tilde{\delta}_t(k) - \log(a_{ki})\Big) - \log(b_i(o_{t+1})).\end{aligned}$$

Natuurlijk worden de logaritmen van de a_{ij} en $b_i(k)$ niet steeds opnieuw berekend, maar ze worden bij het HMM opgeslaan.

> Een soortgelijke opmerking geldt natuurlijk ook voor het forward algoritme. Daarbij is er echter het probleem, dat de kansen voor de verschillende paden bij elkaar opgeteld moeten worden. Dit lost men soms met behulp van de formule $\log(p+q) = \log(p(1+\frac{q}{p})) = \log(p)+\log(1+\frac{q}{p}) = \log(p) + \log(1+e^{\log(q)-\log(p)})$ op, maar vaak wordt hier inderdaad met kansen gerekend die op een geschikte manier geschaald worden.
>
> Ook dit probleem wordt vermeden, als men bij het evalueren het forward algoritme door de Viterbi benadering vervangt.

Toepassing van het Viterbi algoritme

We kijken nu naar de toepassing van het Viterbi algoritme op een HMM met de drie munten, waarvan maar één eerlijk is. De drie munten zijn de drie states S_1, S_2, S_3 en de mogelijke uitkomsten zijn $x_1 = \texttt{K}$ voor *kop* en $x_2 = \texttt{M}$ voor *munt*. Het HMM $\lambda = \lambda(A, B, \pi)$ is gegeven door

$$A = (a_{ij}) := \begin{pmatrix} 0.6 & 0.2 & 0.2\\ 0.4 & 0.2 & 0.4\\ 0.4 & 0.4 & 0.2\end{pmatrix},\quad B = (b_i(k)) := \begin{pmatrix} 0.5 & 0.5\\ 0.75 & 0.25\\ 0.25 & 0.75\end{pmatrix},\quad \pi = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3}).$$

We bekijken de waarneming $O = \texttt{KMKMM}$.

Voor de initialisering hebben we:

$$\begin{aligned}\delta_1(1) &= \pi(1)b_1(1) = 0.33 \cdot 0.5 = 0.167,\\ \delta_1(2) &= \pi(2)b_2(1) = 0.33 \cdot 0.75 = 0.25,\\ \delta_1(3) &= \pi(3)b_3(1) = 0.33 \cdot 0.25 = 0.083.\end{aligned}$$

Voor de volgende stap berekenen we nu

$$\begin{aligned}
i=1: \delta_1(1)a_{11}b_1(2) &= 0.167\cdot 0.6\cdot 0.5 = 0.05, \leftarrow \max\\
\delta_1(2)a_{21}b_1(2) &= 0.25\cdot 0.4\cdot 0.5 = 0.05,\\
\delta_1(3)a_{31}b_1(2) &= 0.083\cdot 0.4\cdot 0.5 = 0.0167,\\
i=2: \delta_1(1)a_{12}b_2(2) &= 0.167\cdot 0.2\cdot 0.25 = 0.0083,\\
\delta_1(2)a_{22}b_2(2) &= 0.25\cdot 0.2\cdot 0.25 = 0.0125, \leftarrow \max\\
\delta_1(3)a_{32}b_2(2) &= 0.083\cdot 0.4\cdot 0.25 = 0.0083,\\
i=3: \delta_1(1)a_{13}b_3(2) &= 0.167\cdot 0.2\cdot 0.75 = 0.025,\\
\delta_1(2)a_{23}b_3(2) &= 0.25\cdot 0.4\cdot 0.75 = 0.075, \leftarrow \max\\
\delta_1(3)a_{33}b_3(2) &= 0.083\cdot 0.2\cdot 0.75 = 0.0125.
\end{aligned}$$

Dit geeft voor de $\delta_2(i)$ het volgende:

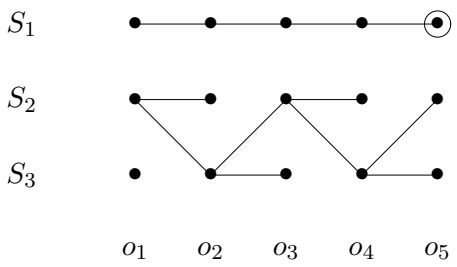
$$\begin{aligned}
&\delta_2(1) = 0.05 \text{ met } k=1 \text{ (of } k=2) \text{ als voorganger},\\
&\delta_2(2) = 0.0125 \text{ met } k=2 \text{ als voorganger},\\
&\delta_2(3) = 0.075 \text{ met } k=2 \text{ als voorganger}.
\end{aligned}$$

Als we zo doorgaan krijgen we voor $\delta_t(i)$ met de voorgangers k:

$$\begin{aligned}
&\delta_3(1) = 0.015, k=1, \quad \delta_3(2) = 0.0225, k=3, \quad \delta_3(3) = 0.00375, k=3,\\
&\delta_4(1) = 0.0045, k=1, \ \delta_4(2) = 0.001125, k=2, \ \delta_4(3) = 0.00675, k=2,\\
&\delta_5(1) = 0.00135, k=1, \delta_5(2) = 0.000675, k=3, \delta_5(3) = 0.0010125, k=3.
\end{aligned}$$

We zien dat $\delta_5(1)$ het maximum van de $\delta_5(i)$ is, daarom eindigt de optimale rij van states in state S_1. Omdat in alle stappen de state S_1 voorganger S_1 heeft, is dus $S_1S_1S_1S_1S_1$ de optimale rij van states. Merk op dat tot $t=4$ de rij $S_2S_3S_2S_3$ optimaal was geweest.

Als we de punten (t,i) als punten van een tralie (of rooster) bekijken en het punt (t,i) met degene voorganger $(t-1,k)$ verbinden die de maximale waarde van $\delta_t(i)$ oplevert, kunnen we hieruit de optimale rij van states makkelijk achterhalen. In Figuur III.7 is dit tralie voor het net besproken voorbeeld te zien, waarbij de optimale eindstate door een extra cirkel benadrukt is.



Figuur III.7: Tralie voor het Viterbi algoritme.

13.3 Training van een HMM

Tot nu toe zijn we ervan uit gegaan dat we de parameters van het HMM al kennen. De vraag is nu, hoe we de parameters $A = (a_{ij})$, $B = (b_i(k))$ en $\pi = (\pi(1), \ldots, \pi(N))$ zo kunnen bepalen, dat het model een gegeven rij $O = o_1 o_2 \ldots o_T$ van waarnemingen zo goed mogelijk beschrijft, dus zo dat de kans $p(O \mid \lambda(A, B, \pi))$ maximaal wordt. Omdat bij deze aanpak de kans gemaximaliseerd wordt, noemt men dit ook de *maximum likelihood schatting* van de parameters.

In Wiskunde 1 hebben we in het kader van de kansrekening naar een soortgelijk, maar veel eenvoudiger probleem gekeken. We wilden toen de parameters van een kansverdeling, bijvoorbeeld een normale verdeling, zo bepalen, dat met deze parameters de kans voor een rij gebeurtenissen maximaal werd. Het idee was toen, de (logaritme van de) kans op de gebeurtenissen als functie van de parameters te interpreteren en een maximum van deze functie te bepalen door de partiële afgeleiden naar de parameters gelijk aan 0 te zetten en deze vergelijkingen op te lossen. Bij de normale verdeling hebben we zo bijvoorbeeld geconcludeerd, dat de beste keuze voor de verwachtingswaarde μ van de normale verdeling het gemiddelde van de gebeurtenissen is – een niet echt verrassend resultaat.

In principe zouden we bij de HMM's een analoge aanpak kunnen kiezen: We schrijven $p(O \mid \lambda)$ als functie van de parameters a_{ij}, $b_i(k)$ en $\pi(i)$, zo als we dat in het begin van deze les al hebben gedaan, dus als

$$p(O \mid \lambda) = \sum_{q=q_1 \ldots q_T} \pi(q_1) b_{q_1}(o_1) a_{q_1 q_2} b_{q_2}(o_2) \ldots a_{q_{T-1} q_T} b_{q_T}(o_T).$$

Vervolgens bepalen we de partiële afgeleiden naar de parameters en proberen de vergelijkingen

$$\frac{\partial}{\partial a_{ij}} p(O \mid \lambda) = 0, \quad \frac{\partial}{\partial b_i(k)} p(O \mid \lambda) = 0, \quad \frac{\partial}{\partial \pi(i)} p(O \mid \lambda) = 0$$

simultaan op te lossen. Dat we eigenlijk nog moeten eisen dat de rijen van de matrices A en B de som 1 hebben, omdat we het over kansverdelingen hebben, vergeten we hierbij even.

Het probleem is dat in alle praktische gevallen het stelsel vergelijkingen dat men zo krijgt niet analytisch oplosbaar is. Maar dit probleem doet zich ook al in veel eenvoudigere vraagstukken voor, want ook bij gewone functies van één veranderlijke kunnen we vaak de nulpunten niet expliciet bepalen. De gebruikelijke manier, om in deze situatie verder te komen, is een *numerieke benaderingsmethode* toe te passen.

Het idee in het kader van HMM's is, startwaarden voor de parameters A, B en π te gokken en vervolgens de parameters stapsgewijs zo aan te passen, dat in elke stap de *likelihood* $p(O \mid \lambda(A, B, \pi))$ toeneemt.

In het algemeen levert zo'n benaderingsmethode alleen maar een lokaal maximum van de functie $p(O \mid \lambda)$ op, en omdat deze functie zo ingewikkeld

is, is er ook geen goede manier om een globaal maximum te vinden. In de praktijk probeert men een paar verschillende stelsels van startwaarden en kiest vervolgens het beste van de gevonden lokale maxima.

Baum-Welch algoritme

We zullen nu een speciale benaderingsmethode bekijken, die de parameters van een HMM stapsgewijs verbetert, namelijk het *Baum-Welch algoritme*. Deze gebruikt de vooruit- en achteruitkansen $\alpha_t(i)$ en $\beta_t(i)$ die we al bij de evaluatie van het HMM tegen gekomen zijn.

Om de methode goed te kunnen formuleren, hebben we eerst nog twee nieuwe uitdrukkingen nodig, die zekere kansen beschrijven:

De voorwaardelijke kans dat het systeem op tijdstip t in state S_i is, gegeven de volledige rij waarnemingen $O = o_1 o_2 \dots o_T$, noemen we $\gamma_t(i)$. Volgens de relatie $p(A \mid B) = \frac{p(A,B)}{p(B)}$ geldt dan:

$$\gamma_t(i) := p(q_t = S_i \mid O, \lambda) = \frac{p(O, q_t = S_i \mid \lambda)}{p(O \mid \lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)}.$$

Verder definiëren we als $\xi_t(i,j)$ de voorwaardelijke kans dat het systeem tussen de tijdstippen t en $t+1$ van state S_i naar state S_j gaat, gegeven de rij O van waarnemingen. Er geldt

$$\begin{aligned}\xi_t(i,j) := p(q_t = S_i, q_{t+1} = S_j \mid O, \lambda) &= \frac{p(O, q_t = S_i, q_{t+1} = S_j \mid \lambda)}{p(O \mid \lambda)} \\ &= \frac{\alpha_t(i)\ a_{ij}\ b_j(o_{t+1})\ \beta_{t+1}(j)}{p(O \mid \lambda)}.\end{aligned}$$

Tussen de kansen $\xi_t(i,j)$ en $\gamma_t(i)$ bestaat een eenvoudige relatie, want de kans om op tijdstip t in state S_i te zijn is de som over alle j van de kansen, tussen de tijdstippen t en $t+1$ van state S_i naar S_j te gaan. Er geldt dus

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j).$$

Als we nu de kansen $\gamma_t(i)$ over de tijdstippen $t = 1, \dots, T$ optellen, krijgen we het verwachte aantal van waarnemingen die door de state S_i geproduceerd zijn. Net zo kunnen we de kansen $\xi_t(i,j)$ over de tijdstippen $t = 1, \dots, T-1$ optellen, dan krijgen we het verwachte aantal overgangen van state S_i naar state S_j. We hebben dus

$$\begin{aligned}\sum_{t=1}^{T} \gamma_t(i) &= \text{ verwacht aantal emissies vanuit state } S_i; \\ \sum_{t=1}^{T-1} \xi_t(i,j) &= \text{ verwacht aantal overgangen tussen states } S_i \text{ en } S_j.\end{aligned}$$

Maar aan de hand van deze gegevens kunnen we nieuwe parameters A', B' en π' als relatieve frequenties schatten, namelijk door:

$$\begin{aligned}
\pi'(i) &= \text{ verwachte kans op state } S_i \text{ op tijdstip 1} \\
&= \gamma_1(i) = \frac{\alpha_1(i)\ \beta_1(i)}{\sum_{i=1}^{N} \alpha_t(i)\ \beta_t(i)} = \frac{\alpha_1(i)\ \beta_1(i)}{\sum_{i=1}^{N} \alpha_T(i)} \\
a'_{ij} &= \frac{\text{verwacht aantal overgangen van state } S_i \text{ naar state } S_j}{\text{verwacht aantal overgangen vanuit state } S_i} \\
&= \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} = \frac{\sum_{t=1}^{T-1} \alpha_t(i)\ a_{ij}\ b_j(o_{t+1})\ \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i)\beta_t(i)} \\
b'_i(k) &= \frac{\text{verwacht aantal emissies vanuit state } S_i \text{ met waarneming } x_k}{\text{verwacht aantal emissies vanuit state } S_i} \\
&= \frac{\sum_{t=1,o_t=x_k}^{T} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)} = \frac{\sum_{t=1,o_t=x_k}^{T} \alpha_t(i)\ \beta_t(i)}{\sum_{t=1}^{T} \alpha_t(i)\ \beta_t(i)}
\end{aligned}$$

Merk op hoe de waarneming $O = o_1 o_2 \dots o_T$ bij de berekening van a'_{ij} en $b'_i(k)$ betrokken is, dit is uiteindelijk de reden dat de parameters aan de waarneming aangepast worden.

De grap is nu, dat we met de nieuwe parameters $A' = (a'_{ij})$, $B' = (b'_i(k))$ en $\pi' = (\pi'(1), \dots \pi'(N))$ steeds een *beter* model voor de beschrijving van O krijgen dan met de oude parameters A, B en π, er laat zich namelijk aantonen dat geldt:

$$\lambda' = \lambda(A', B', \pi') \Rightarrow p(O \mid \lambda') \geq p(O \mid \lambda).$$

We kunnen nu de herschatting van de parameters itereren door het nieuwe model $\lambda(A', B', \pi')$ te gebruiken om de vooruit- en achteruitkansen $\alpha_t(i)$ en $\beta_t(i)$ en de kansen $\gamma_t(i)$ en $\xi_t(i,j)$ opnieuw te bepalen en hieruit een verder verbeterd stelsel parameters te verkrijgen. Deze procedure wordt herhaald tot dat de likelihood $p(O \mid \lambda)$ niet meer veranderd of een maximaal aantal iteratie stappen bereikt is.

13.4 Toegift: Levenshtein afstand

Als toegift behandelen we de toepassing van Bellman's principe op een ander belangrijk probleem in de patroonherkenning, namelijk de afstand tussen strings. Dit heeft toepassingen in de verwerking en herkenning van teksten en taal, maar ook in de beeldherkenning.

Een string is hierbij algemeen een keten van symbolen en men wil een afstand tussen twee ketens kunnen berekenen. Bij teksten zijn de symbolen gewoon letters, in de spraakherkenning zijn de symbolen vaak woorden, maar kunnen ook grammaticale etiketten zijn. In de beeldherkenning wordt vaak de omtrek van een element als keten van zekere elementaire symbolen beschreven, lijnstukken, hoeken etc.

Een mogelijke definitie van de afstand tussen twee strings is de *Edit afstand* die naar een van de uitvinders nu meestal *Levenshtein afstand* heet. Het

idee hierbij is, door *elementaire edit operaties* de ene string in de andere te transformeren, waarbij elementaire operaties de volgende zijn:

- vervangen (substitution) van een symbool, bijvoorbeeld *ki***j***ker* → *ki***k***ker*;
- invoegen (insertion) van een symbool, bijvoorbeeld *bouwer* → *b***r***ouwer*.
- weglaten (deletion) van een symbool, bijvoorbeeld *koe***k** → *koe*;

Natuurlijk zijn er verschillende manieren, om van een string door een combinatie van vervangen, invoegen en weglaten naar een andere string te komen, maar het is voor de hand liggend het minimale aantal stappen als edit afstand tussen de strings te definiëren:

Definitie: De *Levenshtein afstand* tussen twee strings is gedefinieerd als het *minimale aantal* van elementaire edit operaties waarmee de eerste string in de tweede string getransformeerd kan worden.

De vraag is nu hoe men het minimale aantal operaties vindt. Dit gebeurt analoog met het Viterbi algoritme door de methode van het dynamische programmeren.

Het idee is dat men voor twee strings $X = x_1x_2 \dots x_N$ en $Y = y_1y_2 \dots y_M$ stapsgewijs kijkt hoe men beginstukken van de twee strings in elkaar kan transformeren. Volgens Bellman's principe hoeft men hierbij alleen maar het minimale aantal operaties op te slaan om van het beginstuk $x_1 \dots x_i$ van lengte i van X naar het beginstuk $y_1 \dots y_j$ van lengte j van Y te komen. Men krijgt zo een rooster van punten (i, j) voor $0 \leq i \leq N$ en $0 \leq j \leq M$ waarbij we het aantal edit operaties als kosten voor de overgang tussen twee punten interpreteren. In dit geval hebben we (tegenover het Viterbi algoritme) sterke lokale beperkingen, want het punt (i, j) heeft slechts drie mogelijke voorgangers:

(1) het punt $(i-1, j-1)$: In het geval $x_i = y_j$ heeft de overgang van $(i-1, j-1)$ naar (i, j) kosten 0, anders kosten 1. Als $x_i \neq y_j$ is deze overgang namelijk het vervangen van x_i door y_j.

(2) het punt $(i, j-1)$: Deze overgang is het invoegen van het symbool y_j en heeft de kosten 1.

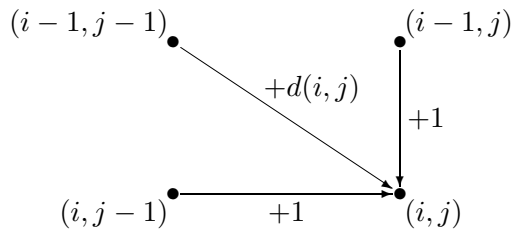
(3) het punt $(i-1, j)$: Deze overgang is het weglaten van het symbool x_i en heeft de kosten 1.

In Figuur III.8 zijn deze overgangen schematisch te zien, waarbij we met $d(i, j)$ de kosten voor het vervangen van x_i door y_j aangeven, hiervoor geldt

$$d(i,j) := \begin{cases} 0 & \text{als} \quad x_i = y_j \\ 1 & \text{als} \quad x_i \neq y_j. \end{cases}$$

Volgens Bellman's principe vinden we de minimale kosten $D(i, j)$ voor de transformatie van het beginstuk $x_1 \dots x_i$ van X naar het beginstuk $y_1 \dots y_j$ van Y als volgt:

We initialiseren $D(i, 0) := i$ voor $0 \leq i \leq N$ (dit is het weglaten van de eerste i symbolen van X) en $D(0, j) := j$ voor $0 \leq j \leq M$ (dit is het invoegen



Figuur III.8: Mogelijke voorgangers van (i,j).

van de eerste j symbolen van Y) en berekenen vervolgens voor $i = 1, 2, \ldots N$ en voor $j = 1, 2, \ldots M$:

$$D(i,j) := \min\{D(i-1,j-1) + d(i,j),\ D(i,j-1) + 1,\ D(i-1,j) + 1\}.$$

Merk op dat op het moment dat we $D(i,j)$ willen berekenen de waarden van $D(i-1,j-1)$, $D(i,j-1)$ en $D(i-1,j)$ al berekend zijn, omdat we i stapsgewijs van 1 t/m N verhogen en voor een vaste i ook met j stapsgewijs van 1 t/m M lopen.

Als we ons de waarden van $D(i,j)$ als elementen van een $N \times M$-matrix voorstellen, vullen we deze matrix rijsgewijs van boven naar beneden en de rijen van links naar rechts. Uiteindelijk zijn we geïnteresseerd in de waarde $D(N,M)$ rechts onder, die de Levenshtein afstand tussen X en Y aangeeft.

Het schema hieronder laat voor het voorbeeld $X =$ `KUNSTMATIGE` en $Y =$ `INTELLIGENTIE` de waarden $D(i,j)$ en een optimaal pad (aangeduid door de hokjes) zien.

		I	N	T	E	L	L	I	G	E	N	T	I	E
	0	1	2	3	4	5	6	7	8	9	10	11	12	13
K	**1**	1	2	3	4	5	6	7	8	9	10	11	12	13
U	2	**2**	3	3	4	5	6	7	8	9	10	11	12	13
N	3	3	**2**	3	4	5	6	7	8	9	9	10	11	12
S	4	4	**3**	3	4	5	6	7	8	9	10	10	11	12
T	5	5	4	**3**	4	5	6	7	8	9	10	10	11	12
M	6	6	5	4	**4**	5	6	7	8	9	10	11	11	12
A	7	7	6	5	5	**5**	6	7	8	9	10	11	12	12
T	8	8	7	6	6	6	**6**	7	8	9	10	10	11	12
I	9	8	8	7	7	7	7	**6**	7	8	9	10	10	11
G	10	9	9	8	8	8	8	7	**6**	**7**	**8**	**9**	**10**	11
E	11	10	10	9	8	9	9	8	7	6	7	8	9	**10**

Merk op dat er verschillende mogelijkheden voor het optimale pad zijn die mogelijk ook verschillende aantallen vervangingen, invoegingen en weglatingen kunnen hebben, maar de *som* van de aantallen vervangingen, invoegingen en weglatingen is bij alle optimale paden natuurlijk hetzelfde. In het voorbeeld is de Levenshtein afstand tussen de twee strings dus 10, het aangegeven pad heeft 4 vervangingen, 4 invoegingen en 2 weglatingen.

Net als bij het Viterbi algoritme moeten we ook hier opslaan vanuit welke voorganger we bij $D(i,j)$ het minimum bereiken om het optimale pad terug te kunnen vinden.

Een iets algemenere versie van de Levenshtein afstand krijgt men, door gewichten aan de verschillende edit operaties te geven, want in sommige toepassingen kan een invoeging erger zijn dan een vervanging. Als we de kosten van een vervanging met k_s, de kosten van een invoeging met k_i en de kosten van een weglating met k_d noteren, berekenen we in dit geval de kosten $D(i,j)$ voor het optimale pad door het punt (i,j) als

$$D(i,j) := \min\{D(i-1,j-1) + d(i,j)\,k_s,\ D(i,j-1) + k_i,\ D(i-1,j) + k_d\},$$

waarbij de initialiseringen $D(i,0) = i\,k_d$ en $D(0,j) = j\,k_i$ zijn.

> In de eerste fase van de spraakherkenning is een soortgelijke techniek ook op spraaksignalen toegepast, er werden namelijk de geluidssignalen in een keten van symbolen omgezet en deze werden door een variatie van de tijdschaal met opgeslagen patronen vergeleken. Deze methode noemt men *dynamic time warping*.

Belangrijke begrippen in deze les

- forward algoritme, backward algoritme
- vooruitkansen, achteruitkansen
- optimale rij van states
- Bellman's principe
- Viterbi algoritme
- training van een HMM, Baum-Welch algoritme
- Levenshtein afstand

Opgaven

99. We beschrijven twee mogelijke uitkomsten `K` en `M` door twee HMM's λ_1, λ_2 met (telkens) twee states. De beginverdelingen voor de states zijn bij beide modellen uniform, dus $\pi = (0.5, 0.5)$. Het model λ_1 heeft de overgangskansen A_1 en emissiekansen B_1, het model λ_2 de overgangskansen A_2 en emissiekansen B_2 gegeven door:

$$A_1 := \begin{pmatrix} 0.6 & 0.4 \\ 0.4 & 0.6 \end{pmatrix},\ B_1 := \begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix};\quad A_2 := \begin{pmatrix} 0.1 & 0.9 \\ 0.9 & 0.1 \end{pmatrix},\ B_2 := \begin{pmatrix} 0.55 & 0.45 \\ 0.45 & 0.55 \end{pmatrix}.$$

(i) Bepaal voor beide modellen de kansen $p(O \mid \lambda)$ voor de waarnemingen $O_1 =$ `KKK` en $O_2 =$ `MKM`.

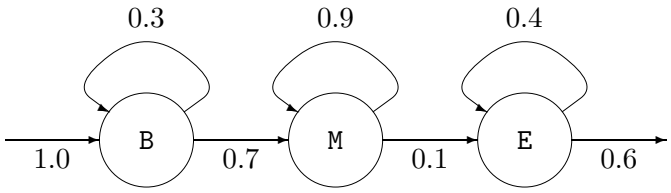
(ii) Bepaal voor beide modellen de optimale rij q van states voor de waarnemingen uit deel (i) en bereken de kansen $p(O, q \mid \lambda)$ voor de combinatie van waarnemingen en states.

100. We kijken nog eens naar het inmiddels bekende HMM met drie munten en parameters:

$$A = (a_{ij}) := \begin{pmatrix} 0.6 & 0.2 & 0.2 \\ 0.4 & 0.2 & 0.4 \\ 0.4 & 0.4 & 0.2 \end{pmatrix}, \quad B = (b_i(k)) := \begin{pmatrix} 0.5 & 0.5 \\ 0.75 & 0.25 \\ 0.25 & 0.75 \end{pmatrix}, \quad \pi = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3}).$$

Door een meting weten we, dat bij de eerste en laatste waarneming de eerlijke (eerste) munt geworpen werd. Wat is nu de optimale rij van states die de waarneming $O =$ `KMKMK` voortbrengt?

101. In de spraakherkenning worden fonemen (de kleinste onderscheidbare klanken in een taal) vaak door HMM's met drie states (begin (`B`), midden (`M`), eind (`E`)) gerepresenteerd. Stel de overgangskansen tussen de states zijn door het volgende diagram gegeven:



Het HMM heeft 7 mogelijke uitkomsten $x_1, \ldots, x_7$ en de emissiekansen voor deze uitkomsten zijn gegeven door

state	x_1	x_2	x_3	x_4	x_5	x_6	x_7
`B`	0.5	0.2	0.3	0	0	0	0
`M`	0	0	0.2	0.7	0.1	0	0
`E`	0	0	0	0	0.1	0.5	0.4

Bepaal voor de rij $x_1x_2x_3x_4x_5x_6x_7$ van uitkomsten de optimale rij van states en geef de kans op deze waarneming voor de optimale rij van states aan.

102. Bepaal de Levenshtein afstand tussen de volgende paren van strings (waarbij ook de spatie een symbool is) en geef de edit operaties aan:

(i) $X =$ `ABABAA` en $Y =$ `ABBAA`;

(ii) $X =$ `IK WEET NIETS` en $Y =$ `WEET IK WAT`;

(iii) $X =$ `SINTERKLAAS` en $Y =$ `KERSTMAN`;

(iv) $X =$ `C3POR2D2` en $Y =$ `HAL2001`.