Tietojenkäsittelytiede **32** Heinäkuu 2011 sivut 48–63
Toimittaja: Antti Valmari

Yksinkertainen on kaunista: Okkamin partaveitsi tilastollisessa mallinnuksessa

Teemu Roos
Tietotekniikan tutkimuslaitos HIIT
Tietojenkäsittelytieteen laitos, Helsingin yliopisto
teemu.roos@cs.helsinki.fi

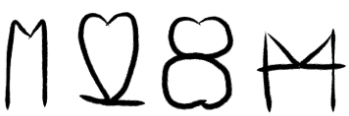
Tiivistelmä

Yksinkertaisuus on vahva induktiivisen päättelyn periaate. Se on läsnä monessa arkielämän tilanteessa epäformaalina peukalosääntönä, jonka mukaan yksinkertaisin selitys on paras. Yksinkertaisuuden periaatetta, eli Okkamin partaveistä, voidaan soveltaa myös tilastollisen päättelyn pohjana. Sen formaali versio, niin sanottu lyhimmän kuvauspituuden periaate (MDL-periaate), asettaa vaihtoehtoiset hypoteesit paremmuusjärjestykseen sen mukaan, mikä niistä mahdollistaa aineiston lyhimmän kuvauksen, kun kuvaus sisältää myös itse hypoteesin. Kuvauspituuden määrittämiseksi sovelletaan informaatioteorian ja tiedon tiivistämisen menetelmiä. Esitän tässä kirjoituksessa joitakin informaatioteorian käsitteitä. Kirjoituksen jälkipuoliskolla käydään läpi MDL-periaatteen alkeita.

1 Johdanto

Elektroniseen pankkiasiointiin liittyvän salaliittoteorian mukaan salainen järjestö vehkeilee suistaakseen maailmanlaajuisen talouselämän ja koko maailman raiteiltaan[1]. Suunnitelman ensimmäinen askel, jalometalleihin perustuvan valuutan korvaaminen virtuaalirahalla, on jo enimmäkseen otettu. Seuraavaksi pankkien tilitiedot aiotaan nollata aiheuttamalla täydellinen sähkökatkos. Kuten salaliittoteorian piirteisiin kuuluu, tätäkään ei voi kieltää tai osoittaa vääräksi joutumatta itse syytetyksi siihen kuulumisesta. Tällä perusteella salaliiton piiriin voikin laskea kuuluvan varsin vaikutusvaltaista väkeä! Samantyyppisiä teorioita on sepitetty mm. UFOista, Apollo-kuulaskeutumisista, New Yorkin terroriskuista, Raamatusta, Elviksestä ja ilmastonmuutoksesta.

Miksi salaliittoteoriat ovat niin suosittuja? Ajatellaan seuraavaa vertausta. Alla oleva kuva esittää tyypillistä ongelmaa ÄO-testissä: mikä tulee seuraavaksi?



(Oikean vastauksen voi katsoa kirjoituksen lopusta.) Sarjan jatkamiseksi voimme puntaroida erilaisia selityksiä, jotka selittävät, miksi neljä ensimmäistä merkkiä näyttävät sellaisilta kuin näyttävät ja ennustavat, millainen seuraava on. Vastaus ei ole selvä, ennen kuin ratkaisu tulee vas-

[1] Katso `http://en.wikipedia.org/wiki/List_of_conspiracy_theories`.

taan, minkä jälkeen se on niin ilmiselvä, ettei kukaan sitä voi epäillä. Vastaavasti jatkaaksemme lukujonoa

$$1, 2, 4, 8, 16, \ldots \quad (1)$$

keksimme helposti eksponentiaalisen jonon $2^0, 2^1, 2^2, 2^3, 2^4, \ldots$, jonka seuraava alkio on $2^5 = 32$. Olemme tähän ratkaisuun niin tyytyväisiä, ettemme edes pohdi vaihtoehtoja. Mutta eikö ole *mahdollista*, että ensin mainittu jono esittää ulkoavaruuden olentojen aakkosia? Eikö ole *mahdollista*, että jono (1) on 4-Stöhr-jono[2], jonka seuraava alkio on 31, ei 32?

Mitä tekemistä ÄO-testeillä ja salaliittoteorioilla on keskenään? Molempiin voidaan soveltaa *Okkamin partaveistä*. Okkamin partaveitsen — tai ajattelun ekonomian tai yksinkertaisuuden periaatteen — mukaan, muiden asioiden ollessa yhtäläiset, yksinkertainen selitys on parempi kuin monimutkainen. ÄO-testien tapauksessa suosimme automaattisesti yksinkertaisinta ratkaisua, joka onkin lähes poikkeuksetta ollut myös testin tekijän mielessä. Avaruusolentohypoteesi puolestaan on huuhaata juuri siksi, *ettei sitä voi mitenkään sulkea pois laskuista*: miten hyvänsä jono jatkuukin, voidaan aina väittää, että juuri sellainen on olentojen aakkosto. Salaliittoteoriat ovat sitkeitä täsmälleen samasta syystä[3].

Toisin sanoen, juuri se seikka, että salaliittoteorioita tai lukusarjoja, jotka perustuvat avaruusolentojen merkkikieleen, ei voi todistaa vääräksi, osoittaa etteivät ne ole oikeita ”selityksiä”. Ne eivät sulje pois mitään vaihtoehtoa ja siten niillä ei ole lainkaan ennustearvoa.

Tietenkään tämä ei merkitse sitä, että yksinkertainen selitys on välttämättä totta ja monimutkainen väärässä. Jos kaikki yksinkertaiset selitykset osoittautuvat vääriksi ja jäljelle jää vain monimutkaisia, jonkin niistä on oltava totta. Okkamin partaveitsi on tässä mielessä *heuristiikka*, peukalosääntö, joka ei voi olla “oikeassa” tai “väärässä”. Sen sijaan se voi olla “hyödyllinen” tai “hyödytön”, riippuen siitä, johtaako se hyödyllisiin johtopäätöksiin, kun sitä sovelletaan erilaisissa tilanteissa.

Kirjoituksessaan *Simplicity: Views of Some Nobel Laureates in Economic Science*, Michael McAleer kertoo vastauksista, jotka hän sai kysyttyään useilta taloustieteen Nobel-palkinnon saajilta, mikä asema yksinkertaisuudella on heidän työssään [16]. Yksi vastaajista, John F. Nash (s. 1928, Nobel 1994), kirjoitti:

> *Kyllä, olen ehdottomasti pitänyt yksinkertaisuutta arvossa, mikä omalla kohdallani näkyy selvästi taloustieteen teoriassa [...]*
>
> *Hyviä esimerkkejä taloustieteellisessä tai siihen liittyvässä teoriassa ovat omat neuvotteluaksioomani ja Shapleyn aksioomat, jotka määrittävät ”Shapleyn arvon”. [...]*

Kaikesta huolimatta, kuten monet muutkin vastaajat, Robert M. Solow (s. 1924, Nobel 1987) huomautti, että yksinkertaisuudesta ei pidä tehdä liian suurta numeroa:

> *Olen selvästi sitä mieltä, että yksinkertaisuus on mallilta toivottava ominaisuus. Tarvitaan kuitenkin täsmennys. Olen valmis uskomaan, että jotkin asiat, joita haluamme*

[2] h-Stöhr-jono määritellään seuraavasti: Olkoon $a_1 = 1$, ja olkoon kokonaisluvulle $n \geq 1$, a_{n+1} pienin kokonaisluku, jota ei voi esittää korkeintaan h:n erisuuren termin summana, jonka jokainen termi kuuluu joukkoon $\{a_1, \ldots, a_n\}$; katso `http://mathworld.worlfram.com/StoehrSequence.html`.

[3] Karl E. Popperin (1902–1994) mukaan teoria on *tieteellinen* vain, jos sen voi falsifioida eli osoittaa vääräksi jollakin mahdollisella evidenssillä [7]. Popper hylkäsi tällä perusteella mm. psykoanalyysin ja marxismin.

> *mallintaa, ovat luontaisesti monimutkaisia eivätkä alistu yksinkertaisuuden edessä. Sellaisissa tapauksissa olisi hölmöä vaatia yksinkertaisuutta.*

Myös Nash toppuuttelee, kun hän kirjoittaa oman vastauksensa lopuksi:

> *On ehdottomasti totta, että yksinkertaisuudella on merkittävä tehtävä, mutta on myös vaikea kuvitella, että voitaisiin asettaa yksinkertainen "yksinkertaisuuden sääntö", jota suoraviivaisesti soveltamalla hyvän tieteellisen tutkimuksen tekeminen olisi helppoa!*

Aion tässä kirjoituksessa raapaista pintaa kysymyksestä, onko Nashin mainitsema "yksinkertaisuuden sääntö" sovellettavissa tilastolliseen mallinnukseen ja jos on, kuinka hyvin se toimii. Tämä raapaisu vie meidät kiertoajelulle informaatioteorian maailmaan, bittien kotikentälle, missä yksinkertaisuutta mitataan entropian, tiedon tiivistämisen ja laskettavuuden käsittein. Sieltä löydämme myös Okkamin partaveitsen nykyaikaisen muodon, *lyhimmän kuvauspituuden periaatteen* (engl. *minimum description length (MDL) principle*).

Vaikka informaatioteoreettiset periaatteet osoittautuvatkin vahvoiksi työkaluiksi mallinnuksessa ja ennustamisessa, joudumme nobelistien tapaan pitämään jäitä hatussa ja muistuttamaan itseämme siitä, että maailmassa todellakin on monta ihmeellistä asiaa, joiden monimutkaisuutta eivät mitkään periaatteet kumoa.

Kirjoituksen alkupuoliskossa tutustutaan informaatioteorian peruskäsitteisiin, erityisesti tiedon pakkaamisen teoreettiseen problematiikkaan. Jälkimmäisessä puoliskossa näitä käsitteitä sovelletaan MDL-periaatteen muotoilussa.

2 Informaatioteoriasta

2.1 Fysikaalinen entropia

Informaatioteorian synty voidaan ajoittaa melkolailla tarkalleen siihen hetkeen, jolloin Claude E. Shannonin (1916–2001) artikkeli *A Mathematical Theory of Communication* julkaistiin vuonna 1948 [13]. Samansuuntaisia ideoita oli kuplinut pinnan alla jo jonkin aikaa, ainakin niistä ajoista, kun Ludvig Boltzmann (1844–1906) keksi kaavan

$$S = k \log W, \tag{2}$$

joka edelleen koristaa hänen hautakiveään (kuva 1). Kaavassa *entropia* S suhteutetaan ideaalikaasun mikrotilojen lukumäärään, W, kun kaasu on tietyssä makrotilassa. Vakio $k = 1,38062 \times 10^{-23}$ joule/kelvin liittyy entropian termodynaamiseen merkitykseen. Olennaista Boltzmannin kaavassa on entropian ja tilojen lukumäärän *logaritminen* suhde[4].

Kaava (2) soveltuu tapauksiin, joissa järjestelmän mikrotiloja voidaan pitää keskenään yhtäläisinä tai yhtä todennäköisinä. Yleistyksen tapaukseen, jossa kullakin tilalla voi olla oma todennäköisyytensä, p_i, esitti J. Willard Gibbs (1839–1903), jonka määritelmän mukaan

$$S = k \sum_i p_i \log \frac{1}{p_i}, \tag{3}$$

missä k on edelleen Boltzmannin vakio. Huomaa että jos mikrotilojen lukumäärä on W ja kukin niistä on yhtä todennäköinen, $p_i = 1/W$ kaikilla i, saadaan sijoittamalla kaavaan (3) tulokseksi

$$k \sum_i \frac{1}{W} \log \frac{1}{1/W} = \frac{kW}{W} \log W = k \log W,$$

eli Gibbsin kaava palautuu Boltzmannin kaavaan.

[4] Tässä kirjoituksessa logaritmifunktion log kantalukuna käytetään kakkosta, jolloin pätee $\log 2^x = x$. Jatkoa varten on hyvä palauttaa mieleen yleiset laskusäännöt $\log xy = \log x + \log y$ ja $\log \frac{1}{x} = -\log x$.

Kuva 1: Boltzmannin hauta Zentralfriedhof-hautausmaalla Wienissä. Alla yksityiskohta kaiverretusta kaavasta. Lähde: Wikipedia.

Shannonin oivallus oli, että entropialla, nyt ilman termodynaamista vakiota,

$$H = \sum_i p_i \log \frac{1}{p_i} \tag{4}$$

on keskeinen asema tiedonsiirron ja tiivistämisen teoriassa. Jotta voimme tarkemmin tutustua Shannonin tuloksiin esitän seuraavaksi joitain koodaamiseen ja sen matematiikkaan liittyviä huomioita. Thomas Cover ja Joy Thomas ovat kirjoittaneet aiheesta kattavan ja helposti lähestyttävän oppikirjan [1].

2.2 Koodaamisen matematiikkaa

Matemaattisista merkinnöistä: Käytän isoja kirjaimia X, Y jne. merkitsemään satunnaismuuttujia ja pieniä kirjaimia x, y jne. merkitsemään niiden arvoja — vaikkakin tästä säännöstä livetään tavan takaa, kun sekaannuksen vaaraa ei (toivottavasti) ole. Arvojoukkoja merkitään kiemuraisemmilla kirjaimilla, jos sellaisia on tarjolla, esim. $\mathcal{X}, \mathcal{Y}$.

Kirjaimet p, q jne. on varattu pistetodennäköisyysfunktioille ja tiheysfunktioille, joihin liittyvä muuttuja merkitään alaindeksillä, esim. p_X, silloin kun se ei asiayhteydestä muuten selviä. Näin ollen merkintä $\Pr[X = x]$ voidaan kirjoittaa muodossa $p_X(x)$ tai pelkästään $p(x)$. Vaihtoehtoisesti todennäköisyyksiä voidaan merkitä kuten kaavoissa (3) ja (4), eli $p_1, p_2, \ldots$. Lausekkeen $\phi(X)$ odotusarvo kirjoitetaan $\mathbb{E}_{X\sim p}[\phi(X)]$, missä alaindeksi ilmaisee satunnaismuuttujan ja sen jakauman. Kun näistä ei ole epäselvyyttä, ne jätetään merkitsemättä.

Datan koodaaminen voidaan formalisoida kuvauksena syötemerkkijonoilta koodimerkkijonoille. Tiedon tiivistämisessä tavoitteena on kuvata syötejonot niin lyhyiksi koodijonoiksi kuin mahdollista, kuitenkin siten, että syötejonot voidaan tarvittaessa palauttaa ennalleen.

Yksinkertaisuuden vuoksi tarkastellaan tapausta, jossa jokainen syötesymboli, $x_1, \ldots, x_n$, koodataan erikseen. Tällaisia koodeja sanotaan *symbolikoodeiksi*.

Kuten on tapana määräämme, että koodien pitää olla bittijonoja. Muut koodaustavat voidaan käytännössä aina esittää binäärikoodilla, joten rajoitus ei ole merkittävä. Näillä oletuksilla koodi on kuvaus $C : \mathcal{X} \to \{0,1\}^*$ syötesymboleilta äärellisille bittijonoille, joita kutsutaan *koodisanoiksi*.

Koodin C *laajennokseksi* (engl. *extension*) sanotaan kuvausta $C^* : \mathcal{X}^* \to \{0,1\}^*$, joka saadaan liimaamalla syötejonoa $x_1,\ldots,x_n$ vastaavat koodisanat peräjälkeen (kuva 2):

$$C^*(x_1,\ldots,x_n) = C(x_1)\ldots C(x_n).$$

Shannonin teoriassa koodin tehokkuutta mitataan kuvittelemalla, että probabilistinen lähde tuottaa satunnaiskoodisanoja, $X_1, X_2, \ldots$, jotka nyt oletetaan yksinkertaisuuden vuoksi riippumattomiksi jakaumalla p. Koodin tehokkuuden mittari on silloin *odotusarvoinen koodinpituus*:

$$\mathbb{E}[\ell(C(X))] = \sum_{x\in\mathcal{X}} p(x)\,\ell(C(x)), \qquad (5)$$

missä $\ell(C(x))$ on koodisanan pituus, kun lähdesymboli on x.

Symbolikoodi C on *dekoodattavissa* (tai *häviötön*), jos sen laajennos, C^*, on injektio eli jos ja vain jos kaikilla $n,m > 0$ pätee

$$(x_1,\ldots,x_n) \neq (y_1,\ldots,y_m) \\ \Rightarrow C^*(x_1,\ldots,x_n) \neq C^*(y_1,\ldots,y_m).$$

Tarkastellaan esimerkiksi seuraavia koodeja:

1. Koodi, jonka koodisanat ovat $\{0,1,10,11\}$, *ei* ole dekoodattavissa: 10 voi tarkoittaa joko 1,0 tai 10.
2. Koodi, jonka koodisanat ovat $\{00,01,10,11\}$, on dekoodattavissa: jokainen bittipari voidaan dekoodata erikseen.
3. Koodi, jonka koodisanat ovat $\{0,01,011,0111\}$, on myös dekoodattavissa. (Mitä tarkoittaa 0011?)

Koodi 3 yllä on dekoodattavissa, mutta aavistuksen epäkäytännöllinen verrattuna vaikkapa koodiin 2. Koodia 3 dekoodattaessa ei nimittäin voida tietää, mitä tarkoittaa koodijono, jonka alku on 0011… ilman, että tiedetään miten koodijono jatkuu: viimeiset kaksi ykköstä voivat yhtä hyvin liittyä joko koodisanaan 011 tai 0111. Sen sijaan koodi 2 on esimerkki ns. *alkuosavapaasta* (engl. *prefix-free*) koodista, tai lyhyemmin alkuosakoodista (*prefix code*). Tällaisessa koodissa mikään koodisana ei ole toisen koodisanan alkuosa.

Alkuosakoodin koodisanojen pituudet $l_1,\ldots,l_n$ toteuttavat tärkeän *Kraftin epäyhtälön* [4]:

$$\sum_{i=1}^{n} 2^{-l_i} \leq 1. \qquad (6)$$

Esimerkiksi koodi 1 yllä ei toteuta Kraftin epäyhtälöä:

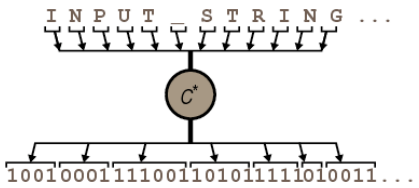
$$2^{-1}+2^{-1}+2^{-2}+2^{-2} \\ = \frac{1}{2}+\frac{1}{2}+\frac{1}{4}+\frac{1}{4} = 1\frac{1}{2} > 1.$$

Sen sijaan koodi 3 toteuttaa:

$$2^{-1}+2^{-2}+2^{-3}+2^{-4} \\ = \frac{1}{2}+\frac{1}{4}+\frac{1}{8}+\frac{1}{16} = \frac{15}{16} \leq 1.$$

Vastaavasti annettuna koodisanojen pituudet $l_1,\ldots,l_n$, jotka toteuttavat Kraftin epäyhtälön, on olemassa alkuosakoodi, jonka koodisanojen pituudet ovat täsmälleen annetut.

Kraftin epäyhtälöä voidaan havainnollistaa seuraavien kaavioiden avulla (kuva 3). Vasemmanpuoleisessa kaaviossa



I	1001
N	0001
P	111001
U	10101
T	1111
_	01
S	0011
⋮	⋮

Kuva 2: Symbolikoodin laajennos C ja esimerkki koodisanoista.

käytettävissä oleva yhden yksikön "budjetti" (kaavion korkeus) riittää kattamaan koodisanoista koituvan "kustannuksen", $\sum_i 2^{-l_i} = 1$. Valitut koodisanat $0, 10, 110, 111$ toteuttavat siis Kraftin epäyhtälön ja itse asiassa koodi on alkuosavapaa. Oikeanpuoleisessa kaaviossa koodisanat ovat $0, 1, 10, 11$. Niiden kustannus, $1\frac{1}{2}$, ylittää budjetin ja Kraftin epäyhtälö ei päde. Koodi ei selvästikään ole alkuosavapaa.

Erittäin hyödyllinen tulos, joka tunnetaan *Kraft–McMillan-teoreemana*, kertoo edellisen pätevän kaikkiin dekoodattaviin koodeihin, ei pelkästään alkuosakoodeihin [6]. Siten rajoittuminen alkuosakoodeihin ei vaikuta millään tavalla saavutettavissa olevaan tehokkuuteen: mikä tahansa dekoodattava koodi voidaan muuntaa vastaavaksi alkuosakoodiksi siten, että koodisanojen pituudet säilyvät ennallaan.

Kraft–McMillan-teoreeman hyödyllisyys liittyy kahteen sen seuraukseen. Ensinnä, kuten sanottu, se osoittaa, ettei rajoittuminen alkuosakoodeihin vaikuta tehokkuuteen. Toinen, vieläkin oleellisempi etu on koodien ja todennäköisyysjakaumien samaistaminen, mikä mahdollistaa probabilististen käsitteiden kuten entropian ja informaation liittämisen koodaamiseen.

Koodien ja jakaumien samaistaminen saavutetaan määrittelemällä mitä tahansa koodinpituuksia $\ell(C(x))$ vastaava todennäköisyysjakauma:

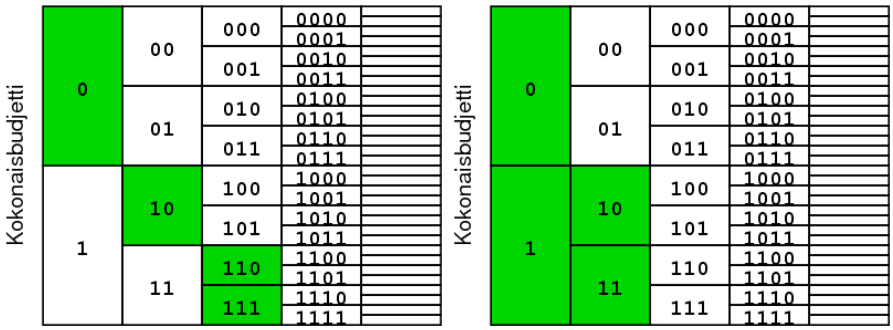
$$q(x) = 2^{-\ell(C(x))} \\ \Leftrightarrow \ \ell(C(x)) = -\log q(x) = \log \frac{1}{q(x)}, \tag{7}$$

kun C on dekoodattava koodi. Koska Kraftin epäyhtälö voi päteä aitona epäyhtälönä, eli koska epäyhtälön vasemman puolen summan arvo saattaa olla vähemmän kuin yksi, määrittää funktio q "*ali*todennäköisyysjakauman", jonka arvojen summa voi olla alle yksi. Näissä tapauksissa koodia voidaan parantaa lyhentämällä ainakin yhtä koodisanaa ilman, että joudutaan ongelmiin. Tällä perusteella kyseessä ei siis ole merkittävä rajoitus, ja ohitamme jatkossa kyseisen kauneusvirheen olettamalla summan olevan tasan yksi.

Kaavan (7) samaistuksella voimme analysoida dekoodattavan koodin tehokkuutta. Odotusarvoinen koodinpituus on nimittäin

$$\mathbb{E}[\ell(C(X))] = \sum_{x \in \mathcal{X}} p(x) \log \frac{1}{q(x)}, \tag{8}$$

mikä muistuttaa selvästi Shannonin entropiaa, kaava (4). Tässä vaiheessa siirrymme tarkastelemaan tarkemmin Shannonin teoriaa, joka tarjoaa mainiot työkalut koodinpituuksien arvioimiseen.



Kuva 3: Kraftin epäyhtälöä havainnollistava kaavio.

2.3 Entropia ja informaatio

Olkoon X satunnaismuuttuja ja p sen pistetodennäköisyysfunktio. Muuttujan arvoon $x \in \mathcal{X}$ liittyvää "yllätystä" voi mitata arvolla

$$I_p(x) = \log \frac{1}{p(x)}.$$

Siis mitä epätodennäköisempi arvo, sitä suurempi yllätys. Muuttujan entropia mittaa *odotusarvoista* yllätystä:

$$H(X) = \mathbb{E}[I_p(X)] = \sum_{x \in \mathcal{X}} p(x) \log \frac{1}{p(x)},$$

joka täsmää Shannonin määritelmän, kaava (4), kanssa.

Kaksiarvoisen satunnaismuuttujan, $X \in \{0,1\}$, entropia on suurin silloin, kun kumpikin arvo on yhtä todennäköinen, katso kuva 4. Varsin loogisesti, kun arvo ei ole lainkaan satunnainen, $\Pr[X=1]=0$ tai $\Pr[X=1]=1$, ei ole odotettavissa lainkaan yllätystä, eli $H(X)=0$.

Toinen tarpeellinen käsite, jonka avulla pääsemme käsiksi koodinpituuksiin, on *Kullback–Leibler-divergenssi* (tai *KL-divergenssi*). Se on määritelty kahden todennäköisyysjakauman, p ja q, funktiona:

$$\begin{aligned} \mathrm{KL}(p\|q) &= \mathbb{E}_{X\sim p}\left[\log \frac{p(X)}{q(X)}\right] \\ &= \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)}. \end{aligned}$$

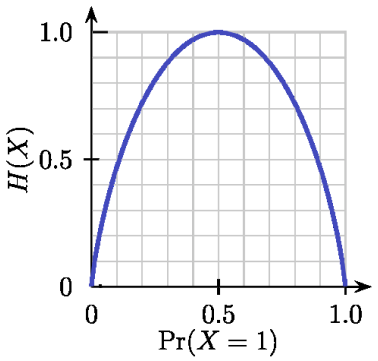
KL-divergenssin tulkinta on seuraava: se antaa odotetun yllätyksen erotuksen, kun käytetään jakaumaa q oikean jakauman p asemesta:

$$\begin{aligned} \mathrm{KL}(p\|q) &= \mathbb{E}_{X\sim p}[I_q(X) - I_p(X)] \\ &= \mathbb{E}_{X\sim p}\left[\log \frac{1}{q(X)} - \log \frac{1}{p(X)}\right]. \end{aligned}$$

Palautetaan nyt mieleen odotusarvoinen koodinpituus, kaava (8), kun koodisanojen pituudet ovat $\ell(C(x)) = \log \frac{1}{q(x)}$ ja syötesymboleita tuotetaan jakaumasta p. Sen voi kirjoittaa myös muodossa:

$$\begin{aligned} \mathbb{E}[\ell(C(X))] &= \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{p(x)q(x)} \\ &= \sum_{x \in \mathcal{X}} p(x) \left[\log \frac{1}{p(x)} + \log \frac{p(x)}{q(x)}\right] \\ &= H(X) + \mathrm{KL}(p\|q), \end{aligned}$$

missä ensin kerrottiin ja jaettiin logaritmin argumentti suureella $p(x)$, minkä jälkeen havaittiin, että summan voi kirjoittaa entropian ja KL-divergenssin summana.



Kuva 4: Kaksiarvoisen satunnaismuuttujan entropia todennäköisyyden $\Pr[X = 1]$ funktiona. Lähde: Wikipedia.

Perustavaa laatua oleva tulos, josta kiitos kuuluu Gibbsille, osoittaa, että KL-divergenssi on ei-negatiivinen:

$$\mathrm{KL}(p\|q) \geq 0,$$

missä epäyhtälö pätee yhtälönä jos ja vain jos argumentit ovat yhtenevät, eli $p(x) = q(x)$ kaikilla $x \in \mathcal{X}$.

Tämä on merkittävää, koska se tarjoaa menetelmän koodisanojen pituuksien valintaan optimaalisella tavalla. Nimittäin, annettuna jakauma p, on viisainta minimoida divergenssi $\mathrm{KL}(p\|q)$, sillä ensimmäinen termi $H(X)$ riippuu pelkästään syötesymboleita tuottavasta jakaumasta p. Koska KL-divergenssi $\mathrm{KL}(p\|q)$ minimoituu kun $p = q$, saavutetaan optimaalinen tehokkuus silloin, kun

$$\ell(C(x)) = \log \frac{1}{p(x)}$$

kaikilla $x \in \mathcal{X}$. Tämä vastaa hyvin intuitiota: lyhyet koodisanat tulee varata todennäköisimmille symboleille, jolloin epätodennäköisille symboleille joudutaan käyttämään pitempiä koodisanoja. Esimerkiksi symboli, jonka todennäköisyys on $1/2$ saa yksibittisen koodisanan (joko 0 tai 1), kun vastaavasti symboli, jonka todennäköisyys on $1/256$ saa kahdeksanbittisen koodisanan, vaikkapa 00101100.

Edellisessä on kuitenkin hankaluutensa. Ensiksi, ei ole mitään takeita siitä, että ideaalinen koodisanan pituus $\log \frac{1}{p(x)}$ on kokonaisluku. (Lukija voi yrittää kuvitella koodisanaa, jonka pituus on $3/4$ tai 0,123.) Toiseksi, vaikka olisimmekin saaneet valittua koodisanojen pituudet, ei ole triviaalia valita koodisanoja siten, että koodi on alkuosavapaa tai edes dekoodattavissa.

Koodisanojen valintapulmaa on tutkittu laajalti. Varhaiset Shannonin ja Robert M. Fanon (s. 1917) laatimat menetelmät olivat jo lähes optimaalisia sikäli, että ne saavuttivat ylärajan

$$\mathbb{E}[\ell(C(X))] \leq H(X) + 1,$$

eli odotusarvoinen koodinpituus oli yhden bitin sisällä entropian määräämästä alarajasta. Shannon ja Fano eivät kuitenkaan pystyneet ratkaisemaan pähkinää täysin tyydyttävällä tavalla. 1950-luvun alkupuolella MIT:ssa pitämänsä informaatioteorian kurssin yhteydessä Fano esittikin ongelman joukolle jatko-opiskelijoita, joiden joukossa oli David A. Huffman (1925–1999). Huffman ratkaisi ongelman

ja nykyään ratkaisu tunnetaan Huffman-koodina, jota käytetään edelleen lukuisissa eri yhteyksissä tiedon tiivistämiseen.

Myöhemmin on esitetty lukemattomia parannuksia ja muunnelmia, joiden avulla tietoa voidaan tiivistää paremmin ja nopeammin. Koodisanojen pituuteen liittyvä kokonaislukurajoite voidaan kiertää mm. soveltamalla symbolikoodien asemesta *aritmeettista koodia*, jonka kehitti suomalainen informaatioteorian pioneeri Jorma Rissanen [8]. Lisäksi on kehitetty monenlaisiin erityistarpeisiin, kuten kuvien, äänen, videon, tekstin jne. pakkaamiseen hyvin soveltuvia algoritmeja. Niissä voidaan usein tyytyä tiedon likimääräiseen tallentamiseen, jolloin syötettä ei voi rekonstruoida täysin häviöttömästi pakkaamisen jälkeen. Aiheesta kiinnostuneet voivat tutustua esimerkiksi Solomonin kirjaan [14] ja sieltä löytyviin viitteisiin.

3 MDL-periaate

3.1 Yleistä

Tässä kappaleesssa pohdimme, miten edellä kuvattuja tiedon tiivistämiseen kehitettyjä käsitteitä ja menetelmiä voidaan soveltaa tilastolliseen päättelyyn. Erityisesti tutustumme Okkamin partaveitsen nykyaikaiseen formaaliin versioon, eli MDL-periaatteeseen. Verrattuna perinteiseen tilastolliseen päättelyyn tai Bayes-inferenssiin, MDL-periaate on tuore tulokas. Aiheesta lisätietoa haluava voi tutustua erittäin kattavaan, joskin ehkä hiukan raskassoutuiseen Peter D. Grünwaldin kirjoittamaan oppikirjaan [2]. Ytimekkäämpää esitystapaa arvostava löytää sellaista Rissasen jo hiukan vanhentuneesta, mutta sitäkin lukijaystävällisemmästä "pienestä vihreästä kirjasta" [11].

MDL-periaatteen kehitti edellä aritmeettisen koodin yhteydessä mainittu Jorma Rissanen [9]. Rissasen motivaationa olivat klassisen tilastotieteen hankaluudet tapauksissa, joissa verrattavana on malleja, joista kaikkien monimutkaisuus ei ole sama. Ratkaisun inspiraationa toimi myös ns. algoritminen informaatioteoria (Kolmogorov-kompleksisuus) ja sen piirissä *universaalikoodin* käsite, ks. [5], sekä aikaisemmin esitetty lyhimmän viestinpituuden (engl. *minimum message length, MML*) periaate [15].

MDL-periaatteen kannalta kolme keskeistä käsitettä ovat *kompleksisuus*, *informaatio* ja *kohina*. Karkeasti ottaen, niiden suhde on

$$\textit{kompleksisuus} \approx \textit{informaatio} + \textit{kohina}.$$

Tavoitteena on erottaa aineistossa piilevä informaatio häiritsevän kohinan joukosta. Yksinkertaisimmillaan MDL-periaatteen mukaan on valittava hypoteesi, joka minimoi kokonaiskuvauspituuden:

$$\min_{h \in \mathcal{H}} (\ell(h) + \ell(D;h)), \qquad (9)$$

missä $\ell(h)$ on koodinpituus, joka vaaditaan hypoteesin kuvaamiseen, ja $\ell(D;h)$ on datan koodinpituus, kun se koodataan hypoteesin h avulla. Käsittelemme kumpaakin termiä tarkemmin alla.

Informaatioteoreettisten lähestymistapojen, kuten MDL:n, suurin vahvuus on niiden luonteva ja hyvin perusteltu ratkaisu monimutkaisuudeltaan eroavien hypoteesien vertailun ongelmaan. Peruskaavasta (9) on helppo nähdä, että vaikka monimutkainen hypoteesi h_{moni} saavuttaakin lyhyen datan koodinpituuden $\ell(D;h_{\text{moni}})$, voi se jäädä kakkoseksi, kun sitä verrataan yksinkertaisempaan hypoteesiin kokonaiskoodinpituuden perusteella.

Kiinnostava ja käytännössä olennainen kysymys kuuluu, miten määritellään koodinpituudet $\ell(h)$ ja $\ell(D;h)$. Esitän kolme vaihtoehtoista tapaa, joista sopiva riippuu vertailtavien hypoteesien luonteesta.

3.2 Ei-stokastiset hypoteesit

Silloin kun hypoteeseina on esimerkiksi kokonaislukujonojen jatkoa koskevia selityksiä, joista jokainen määrittää täsmälleen yhden jonon, on datan koodinpituus nolla, $\ell(D;h) = 0$, jos havaittu jono sopii yhteen selityksen kanssa. Hypoteesin kuvaamisen jälkeen lukujono nimittäin tunnetaan, eikä sitä tarvitse sen enempää kuvailla. Jos sen sijaan lukujono ei sovi yhteen selityksen kanssa, ajatellaan koodinpituuden olevan ääretön:

$$p(D;h) = 0 \;\Rightarrow\; \log \frac{1}{p(D;h)} = \infty.$$

Annettuna lukujonon muutama ensimmäinen alkio ja kaksi vaihtoehtoista selitystä, joista vain toinen sopii yhteen havaintojen kanssa, valitaan MDL-periaatteen(kin) perusteella hypoteeseista yhteensopiva. Jos taas molemmat hypoteesit sopivat yhteen lukujonon kanssa, valitaan niistä yksinkertaisempi.

Käytännössä ei usein voida odottaa, että hypoteesi selittää aineiston joko täysin yksikäsitteisesti tai ei ollenkaan. Sen sijaan voidaan joutua hyväksymään jonkinasteinen eroavuus tai epätäsmällisyys, joka voidaan ottaa huomioon koodattaessa dataa hypoteesin avulla. Tämän seurauksena datan koodinpituus annettuna hypoteesi ei ole enää nolla tai ääretön. Oletetaan esimerkiksi, että verrataan kahta yhtä monimutkaista hypoteesia, $\ell(h_1) = \ell(h_2)$. Oletetaan lisäksi, että hypoteesi h_1 sopii yhteen vain yhden lukujonon kanssa, mutta hypoteesin h_2 mukaan melkein mikä tahansa lukujono on mahdollinen — voidaan ajatella esimerkiksi lukujonoa, joka esitetään kirjoituksen alussa mainitussa avaruusolentojen aakkostossa. Tällaisessa tilanteessa hypoteesi h_1 tuottaa koodinpituuden $\ell(D;h_1) = 0$, mutta hypoteesilla h_2 vastaava koodinpituus on nollasta poikkeava, $\ell(D;h_2) > 0$. Siten MDL-periaate suosii täsmällisempää hypoteesia h_1.

Erityisen suosittu tapa käsitellä tapauksia, joissa aineiston ei oleteta olevan täysin yhteensopiva hypoteesin kanssa, ovat säännöstä poikkeavien havaintojen koodaamiseen perustuvat koodit. Jos havaittu n:n pituinen bittijono sopii yhteen hypoteesin kanssa k:ssa kohdassa ja eroaa muiden $n-k$ bitin kohdalla, voidaan erotus koodata luettelemalla eroavien bittien indeksit. Suoraviivainen tapa vaatii $(n-k)\lceil \log n \rceil$ bittiä[5] ($\lceil \log n \rceil$ bittiä jokaista $(n-k)$ eroavaisuutta kohti). Lisäksi on tarpeen koodata eroavaisuuksia lukumäärä, joka on kokonaisluku välillä $0,\ldots,n$, joten siihen riittää $\lceil \log(n+1) \rceil$ bittiä. Tarvittaessa on koodattava myös jonon pituus n[6].

Hieman paranneltu menetelmä perustuu huomioon, että tapoja valita $n-k$ eroavaa kohtaa n vaihtoehdosta on

$$\binom{n}{n-k} = \binom{n}{k} = \frac{n!}{k!(n-k)!},$$

missä $n!$ merkitsee luvun n kertomaa. Vaadittava koodinpituus on edellisen logaritmi pyöristettynä ylöspäin:

$$\lceil \log n! - \log k! - \log(n-k)! \rceil. \qquad (10)$$

Itse koodisanat voidaan valita yksinkertaisesti järjestämällä n alkion perusjoukon $n-k$ alkion osajoukot etukäteen sovittuun järjestykseen ja valitsemalla kunkin osajoukon koodisanaksi sen järjestysnumeron binääriesitys vähennettynä yksösellä. Esimerkiksi silloin kun $n = 5$ ja

[5] Merkintä $\lceil \cdot \rceil$ merkitsee ylöspäin pyöristystä. Siten jos indeksit ovat kokonaislukuja välillä $0,\ldots,5$, voidaan kukin niistä koodata $\lceil \log 6 \rceil = 3$ bitillä; vaihtoehtoja on kuusi, joista jokaisen voi yksilöidä kolmebittisellä koodisanalla.

[6] Mielivaltaisten kokonaislukujen koodaamiseksi on olemassa menetelmiä, joita ei tässä tilan puutteen vuoksi voida käsitellä, ks. esim. [10].

$n-k=3$, on osajoukkojen "aakkosjärjestys" seuraava:

$$\{1,2,3\},\{1,2,4\},\{1,2,5\},\{1,3,4\},\{1,3,5\},\{1,4,5\},\{2,3,4\},\{2,3,5\},\{2,4,5\},\{3,4,5\}$$

ja koodisanojen pituudeksi riittää $\lceil \log 5! - \log 2! - \log 3! \rceil = 4$ bittiä. Järjestyksessä viidennen osajoukon $\{1,3,5\}$ koodisana olisi siten luvun 4 binääriesitys 0100, jossa alussa oleva nolla tarvitaan pitämään koodisanat vakiomittaisina ja koodi alkuosakoodina.

Kuva 5 havainnollistaa poikkeuksiin perustuvaa koodia. Hypoteesin mukaan datana on musta neliö, jonka koko on 25×25 pikseliä. Sääntöön poikkeuksia muodostavat valkeat pisteet eri kohdissa neliötä, jotka koodataan antamalla ensin niiden lukumäärä $\lceil \log(n+1) \rceil$ bitin koodisanalla ja koodaamalla sen jälkeen niiden paikat kaavan (10) koodilla. Kokonaiskoodinpituudeksi tulee tällöin

$$\ell(D;h) = \lceil \log(n+1) \rceil + \left\lceil \binom{n}{k} \right\rceil, \quad (11)$$

missä $n = 625$ ja k on valkeiden pisteiden lukumäärä. Tätä koodia voi verrata vaikkapa suoraviivaiseen koodiin, jossa jokaista pikseliä vastaa yksi bitti (0 tarkoittaa mustaa, 1 valkeaa), jolloin koodinpituus on datasta riippumatta 625. Huomaa, että poikkeuksiin perustuva koodi antaa usein huomattavasti tätä lyhyemmän koodinpituuden (kuva 5), vaikkei sentään aina. Olemme siis saaneet aikaan hyvin alkukantaisen kuvanpakkausalgoritmin!

3.3 Stokastiset pistehypoteesit

Sääntöihin ja poikkeuksiin perustuvia hypoteesejakin hyödyllisemmän hypoteesiluokan muodostavat stokastiset hypoteesit eli hypoteesit, jotka vastaavat todennäköisyysjakaumia. Käsittelemme ensin tapausta, jossa kukin hypoteesi on pistehypoteesi eli yksittäinen jakauma. Tässä tapauksessa ratkaisu löytyy suoraan Shannonin kehikosta, jonka mukaan koodinpituus on

$$\ell(D;h) = \log \frac{1}{p_h(D)}, \quad (12)$$

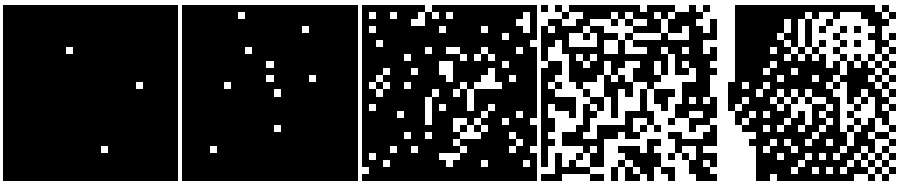
missä $p_h(D)$ on datan todennäköisyys hypoteesin h mukaisessa jakaumassa. Kuten todettu edellä, näin laskettava ideaalinen koodinpituus ei välttämättä ole kokonaisluku. Tämä ei kuitenkaan ole lainkaan ongelmallista MDL-periaatteen kannalta, koska sen sovelluksissa ei koskaan ole tarvetta konstruoida itse koodisanoja; niiden pituuksien tietäminen riittää.

Tässäkin tapauksessa koodinpituus $\ell(h)$ täytyy määrittää huolellisesti. Erityisesti, jos jakaumat kuuluvat perheeseen, johon liittyy jatkuva-arvoisia parametreja, kuten Poisson- tai geometriset jakaumat, on parametrit ensin karkeistettava äärelliseen tarkkuuteen, jotta arvot voidaan koodata äärellisellä määrällä bittejä; katso seuraavan kappaleen kaksiosaisia koodeja käsittelevä kohta.

3.4 Stokastiset komposiittihypoteesit

Todellinen haaste mallinvalintakriteereille ovat ns. komposiittihypoteesit, eli hypoteesit, jotka vastaavat jakaumien joukkoa eli *malliluokkaa*. Voidaan esimerkiksi haluta valita sopiva muuttujaosajoukko, jota käytetään lineaariregressiomallin riippumattomina muuttujina (syötteinä), tai toisaalta voidaan valita aikasarjamallin muistin pituus. Suosittuja menetelmiä, jotka soveltuvat vastaaviin tilanteisiin, ovat Akaiken informaatiokriteeri (AIC), bayesiläinen informaatiokriteeri (BIC) jne.

Jotta MDL-periaatetta voidaan soveltaa komposiittihypoteesien vertailuun,



a) $k = 3$ $\ell(D;h) = 36$ *b*) $k = 10$ $\ell(D;h) = 81$ *c*) $k = 100$ $\ell(D;h) = 402$ *d*) $k = 300$ $\ell(D;h) = 630$ *e*) $k = 372$ $\ell(D;h) = 614$

Kuva 5: Koodi, joka perustuu hypoteesiin "musta neliö, jossa valkeita pisteitä". Koodinpituus lasketaan kaavan (11) avulla, missä $n = 25 \times 25 = 625$ ja k on valkeiden pisteiden lukumäärä.

konstruoidaan jokaisen malliluokan pohjalta *universaalikoodi*. Universaalikoodin tavoite on saavuttaa mille tahansa datalle lähes yhtä lyhyt koodi kuin saavutettaisiin valitsemalla malliluokasta kyseiselle datalle paras jakauma. Koska universaalikoodi antaa kiinteälle datalle yksikäsitteisen koodinpituuden, voidaan sen perusteella verrata mitä tahansa malleja, oli niiden monimutkaisuus mikä tahansa.

Koska jäljempänä esitettävien universaalikoodien käsittely on aiheen monimutkaisuuden vuoksi hiukan tähänastista vaikeaselkoisempaa, on ehkä hyödyllistä ottaa mukaan seuraava yksinkertaistettu esimerkki. Ajatellaan kahta komposiittihypoteesia eli malliluokkaa. Malliluokkaan M_1 liittyy kaksi riippumatonta binäärimuuttujaa X ja Y, joilla kummallakin on oma Bernoulli-jakaumansa. Vastaavat yhteistodennäköisyysjakaumat voidaan esittää muodossa

$$p_{M_1}(x,y;\theta_x,\theta_y) = \begin{cases} (1-\theta_x)(1-\theta_y), & \text{jos } x=0, y=0 \\ (1-\theta_x)\theta_y, & \text{jos } x=0, y=1 \\ \theta_x(1-\theta_y), & \text{jos } x=1, y=0 \\ \theta_x\theta_y, & \text{jos } x=1, y=1, \end{cases} \quad (13)$$

missä θ_x ja θ_y ovat parametreja, joiden arvot ovat välillä $[0,1]$.

Malli M_2 puolestaan sisältää kaikki kahden binäärimuuttujan yhteistodennäköisyysjakaumat, myös sellaiset, joissa X ja Y eivät ole riippumattomia:

$$p_{M_2}(x,y;\theta_{00},\theta_{01},\theta_{10},\theta_{11}) = \begin{cases} \theta_{00}, & \text{jos } x=0, y=0 \\ \theta_{01}, & \text{jos } x=0, y=1 \\ \theta_{10}, & \text{jos } x=1, y=0 \\ \theta_{11}, & \text{jos } x=1, y=1. \end{cases} \quad (14)$$

Parametreja on nyt neljä; tosin kompaktimpi parametrisointi hyödyntää sitä, että $\theta_{00} + \theta_{01} + \theta_{10} + \theta_{11} = 1$, minkä vuoksi malliluokan M_2 dimensio (vapausasteiden lukumäärä) on neljän asemesta kolme. Jälkimmäinen malli on ekvivalentti multinomijakauman kanssa, jossa vaihtoehtoisia arvoja on $k = 4$, yksi kutakin X:n ja Y:n arvojen kombinaatiota kohti.

Kumpikin malli voidaan laajentaa koskemaan n riippumatonta ja samoin jakautunutta havaintoa määrittelemällä

$$p(D;\theta,M) = \prod_{i=1}^{n} p(x_i,y_i;\theta_j,M), \quad (15)$$

missä $D = ((x_1,y_1),\ldots,(x_n,y_n))$ on havaintojono ja θ merkitsee kyseisen mallin, $M \in \{M_1, M_2\}$, kaikkia parametreja.

Tarkastellaan konkreettisuuden vuoksi seuraavaa kymmenen havainnon aineistoa:

	1	2	3	4	5	6	7	8	9	10
x	0	1	1	0	1	0	0	1	0	0
y	0	0	0	1	1	0	1	0	1	1

Seuraavassa esitän universaalimallien kolme päätyyppiä, joista jokaisen kohdalla palataan yllä esitettyyn esimerkkitapaukseen.

1. Kaksiosainen koodi: Historiallisesti varhaisin universaalikoodityyppi perustuu jakaumien parametrien optimaaliseen karkeistukseen, jonka avulla parametrit voidaan koodata äärellisen pituisella koodilla. Tämän jälkeen data koodataan tavalliseen tapaan Shannonin teorian mukaan optimaalisella koodilla, joka määräytyy koodattujen parametrien tuottamasta jakaumasta kaavan (12) tapaan.

Menemättä yksityiskohtiin kokonaiskoodinpituutta voidaan asymptoottisesti arvioida lausekkeella

$$\min_{\theta\in\Theta} \log \frac{1}{p(D;\theta,M)} + \frac{k}{2}\log n,$$

missä parametri θ valitaan parametriavaruudessa Θ, joka määräytyy malliluokan M mukaan, ja k on malliluokan (vapaiden) parametrien lukumäärä. Helposti nähdään, että parametreiksi valikoituvat tutut suurimman uskottavuuden parametrit, $\hat{\theta}(D)$. Siten kaksiosaisen universaalikoodin asymptoottinen pituudeksi saadaan

$$-\log p(D;\hat{\theta}(D),M) + \frac{k}{2}\log n, \qquad (16)$$

mikä vastaa itse asiassa täsmälleen BIC-kriteeriä.

Sovellamme kaavan (16) approksimaatiota yllä esitettyyn esimerkkitapaukseen. Sitä varten tarvitaan molempien mallien suurimman uskottavuuden parametrit, jotka mallin M_1 tapauksessa ovat

$$M_1: \quad \hat{\theta}_x = 4/10, \quad \hat{\theta}_y = 5/10,$$

ja mallin M_2 tapauksessa vastaavasti

$$M_2: \quad \begin{array}{ll} \hat{\theta}_{00} = 2/10, & \hat{\theta}_{01} = 3/10, \\ \hat{\theta}_{10} = 4/10, & \hat{\theta}_{11} = 1/10. \end{array}$$

Soveltamalla kaavoja (16) ja (13)–(15) saadaan nyt koodinpituuksiksi

$$\begin{aligned} &\ell_{\text{approx}}(D;M_1) \\ &= -\log p(D;\hat{\theta}_x,\hat{\theta}_y,M_1) + \frac{2}{2}\log 10 \\ &= -\log[(1-\hat{\theta}_x)^6\hat{\theta}_x^4(1-\hat{\theta}_y)^5\hat{\theta}_y^5] + \log 10 \\ &\approx 19,7 + 3,3 = 23,0, \end{aligned} \qquad (17)$$

ja

$$\begin{aligned} &\ell_{\text{approx}}(D;M_2) \\ &= -\log p(D;\hat{\theta}_{00},\hat{\theta}_{01},\hat{\theta}_{10},\hat{\theta}_{11},M_2) \\ &+ \frac{3}{2}\log 10 \\ &= -\log \hat{\theta}_{00}^2\hat{\theta}_{01}^3\hat{\theta}_{10}^4\hat{\theta}_{11} + \frac{3}{2}\log 10 \\ &\approx 18,5 + 5,0 = 23,4. \end{aligned} \qquad (18)$$

Tällä perusteella pitäisimme täpärästi mallia M_1 parempana. Huomaa, että tämä johtuu mallin M_2 suuremmasta kompleksisuudesta, vaikka se tuottaakin lyhyemmän koodinpituuden datalle (18,5 bittiä) kuin malli M_1 (19,7 bittiä).

2. Sekoitejakaumakoodi: Toinen tapa konstruoida universaalikoodeja perustuu sekoitejakaumiin, joita sovelletaan usein Bayes-päättelyssä. Tällöin koodinpituus on muotoa

$$-\log \int_\Theta p(D;\theta,M)\, w_M(\theta)\, d\theta,$$

missä w_M on parametrien priorijakauma. MDL-periaatteen piirissä prioria sovelletaan teknisenä työkaluna ilman sen bayesiläistä tulkintaa[7].

[7] Rissanen kuvailee asiaa seuraavasti [12]:

Sekoitejakaumakoodiin liittyvät laskutoimitukset voivat vaikuttaa niihin ensi kertaa tutustuvista lukijoista hiukan hankalilta, missä tapauksessa ne voi sivuuttaa ja kiinnittää huomiota pelkästään saataviin koodinpituuksiin, kaavat (19) ja (20).

Jotta voimme soveltaa sekoitejakaumakoodia esimerkkitapauksessa, tarvitsemme siis parametripriorin. Mallille M_1 voidaan käyttää tasaista jakaumaa $\theta \sim \mathrm{Uni}(0,1)$, joka antaa kaikille vakiomittaisille intervalleille välillä $[0,1]$ saman todennäköisyyden. Sovellamme tasajakaumaa molemmille parametreille, θ_x ja θ_y, ja oletamme niiden olevan toisistaan riippumattomia.

Mallin M_2 kohdalla on otettava huomioon rajoite, jonka mukaan parametrien summa on yksi. Rajoitteen toteuttavien parametrivektorien joukossa tasainen jakauma on Dirichlet-jakauma $\mathrm{Dir}(1,1,1,1)$. Koodinpituudet voidaan nyt laskea Bayes-teoriassa tutuista Dirichlet–multinomi-integraaleista, joiden avulla saadaan

$$\begin{aligned}&\ell_{\text{sekoite}}(D;M_1)\\ &= -\log\left(\frac{n_{[x=0]}!n_{[x=1]}!}{n!}\,\frac{n_{[y=0]}!n_{[y=1]}!}{n!}\right)\\ &= -\log\frac{6!4!5!5!}{10!10!} \approx 15,7,\end{aligned} \tag{19}$$

missä $n_{[x=0]}$ on niiden havaintojen lukumäärä, joilla $x=0$ jne.; ja vastaavasti mallille M_2

$$\begin{aligned}&\ell_{\text{sekoite}}(D;M_2)\\ &= -\log\frac{n_{[xy=00]}!n_{[xy=01]}!n_{[xy=10]}!n_{[xy=11]}!}{n!}\\ &= -\log\frac{2!4!3!1!}{10!} \approx 13,6.\end{aligned} \tag{20}$$

Siten sekoitejakaumakoodin perusteella valitaan malli M_2. Huomataan, että valinta eroaa kaksiosaisen koodin approksimaation perusteella tehdystä valinnasta.

3. NML-koodi: Tuorein universaalikoodityyppi perustuu ns. normalisoituun suurimman uskottavuuden jakaumaan (engl. *normalized maximum likelihood, NML*), joka määritellään seuraavasti:

$$p_{\text{nml}}(D;M) = \frac{p(D;\hat{\theta}(D),M)}{C_M}, \tag{21}$$

missä

$$C_M = \sum_{D'\in\mathcal{X}^n} p(D';\hat{\theta}(D'),M)$$

on normalisointivakio, jonka määrittävän summalausekkeen termeinä ovat kaikki mahdolliset datan D kokoiset datajoukot $D' \in \mathcal{X}^n$. Kahden binäärimuuttujan esimerkkitapauksessamme data-avaruus on $\mathcal{X} = \{(0,0),(0,1),(1,0),(1,1)\}$. Huomaa, että summan termien $D' \in \mathcal{X}^n$ lukumäärä kasvaa eksponentiaalisesti suhteessa otoskokoon n.

NML-universaalikoodin koodinpituudeksi saadaan kaavan (21) käänteisluvun logaritmi, eli

$$-\log p(D;\hat{\theta}(D),M) + \log C_M. \tag{22}$$

Jälkimmäisestä termistä $\log C_M$ käytetään nimitystä parametrinen kompleksisuus, koska se kertoo, kuinka paljon ylimääräistä koodinpituutta koituu siitä, ettei optimaalisia parametriarvoja $\hat{\theta}(D)$ tunneta etukäteen. Normalisointitermi C_M voidaan tulkita myös mallin sisältämien *toisistaan erotettavissa olevien* jakaumien lukumääräksi. Koska erotettavuutta ei voi määritellä yksikäsitteisesti, on ymmärrettävää, että lukumääräkään ei tässä tapauksessa ole välttämättä kokonaisluku.

MDL-periaatteen mukaisessa tilastollisessa päättelyssä ei tarvita parametrien priorijakauman epämukavaa bayesiläistä tulkintaa. Sen sijaan voimme käsittää jakaumat, kuten $[p(D;M)]$, luokan mallien konveksisiksi lineaarikombinaatioiksi, joiden hyödyllisyys arvioidaan muilla perusteilla...

Sovellamme nyt NML-koodia esimerkkitapaukseen. Yksityiskohdista vähemmän kiinnostunut lukija voi taas kiinnittää huomionsa pelkästään lopputulokseen, kaavat (23) ja (24).

NML-koodinpituus, kaava (22), koostuu kahdesta osasta, joista ensimmäinen on sama kuin kaksiosaisen koodin approksimaatiossa, kaavoissa (17)–(18). Jälkimmäinen termi on normalisointitermi, joka määritelmän mukaan sisältää summan, jonka suoraviivainen laskeminen edellyttäisi kaikkien mahdollisten $n = 10$ havainnon datajoukkojen läpikäymistä. Summa voidaan onneksi laskea lineaarisessa ajassa käyttäen hyväksi häkellyttävän yksinkertaista rekursiokaavaa [3].

Mallilla M_1 normalisointitermi on tavallisen Bernoulli-mallin vastaavan NML-normalisointitermin neliö, jonka laskeminen on suoraviivaista, ks. esim. [2]:

$$C_{M_1} \approx 4,66^2 = 21,7.$$

Mallilla M_2 normalisointitermi on sama kuin neliarvoisen multinomimuuttujan sisältävän mallin vastaava termi, jonka arvo on

$$C_{M_2} \approx 38,0,$$

ks. [3]. NML-koodinpituuksiksi saadaan siten

$$\begin{aligned} &\ell_{\mathrm{nml}}(D;M_1) \\ &\quad = -\log p(D;\hat{\theta}_x,\hat{\theta}_y,M_1) + \log C_{M_1} \\ &\quad \approx 19,7+4,4 = 24,1, \end{aligned} \tag{23}$$

ja

$$\begin{aligned} &\ell_{\mathrm{nml}}(D;M_2) \\ &\quad = -\log p(D;\hat{\theta}_{00},\hat{\theta}_{01},\hat{\theta}_{10},\hat{\theta}_{11},M_2) \\ &\quad + \log C_{M_2} \\ &\quad \approx 18,5+5,2 = 23,7. \end{aligned} \tag{24}$$

Myös NML-koodi johtaa siis mallin M_2 voittoon, vaikka voittajan löytämiseksi tarvittiinkin taas rangaistuspotkuja.

Edellä kuvattu tilanne on itse asiassa melko tyypillinen: kaksiosaisen koodin approksimaation taipumus rangaista kompleksisia malleja raskaammin kuin sekoitejakauma- tai NML-koodin on tunnettu "fakta" (ts. yleensä näin, joskus toisinpäin). On muistettava myös, että sekoitejakauman tapauksessa tulos olisi voinut olla toinen, jos käytetyt parametripriorit olisivat olleet muunlaiset.

Eri universaalimalleihin pohjautuvien päätelmien eroaminen on luonnollisesti kiusallista. Voidaan kuitenkin osoittaa, että aineiston määrän kasvaessa erojen on tapana kadota ja eri kriteerien johtaa samaan johtopäätökseen. Yksityiskohtia vaativa lukija voi tutustua niihin esimerkiksi lähteen [2] parissa. Toinen kiinnostava pohdinnan aihe on MDL-mallinvalinnan ja bayesiläisessä päättelyssä käytettävien ns. Bayes-faktoreiden yhtyminen sekoitejakaumia käytettäessä. Edellyttäen, että, kuten edellä juuri mainittiin, universaalimallien erot katoavat pitemmän päälle, näyttäisi loogiselta päätellä, että MDL-periaate ja Bayes-päättely eivät juurikaan eroa toisistaan. Tämä on kuitenkin ennenaikainen tuomio, koska elävässä elämässä aineiston määrä on aina rajallinen ja siitä johtuvat erot ovat usein merkittäviä.

4 Lopuksi

Tässä kirjoituksessa on parhaassakin tapauksessa vain raapaistu pintaa informaatioteoriasta ja sen sovelluksista tilastollisessa mallinnuksessa. Lisälukemista MDL-periaatteen teoriasta ja sovelluksista löytyy lähdeluettelossa mainituista teoksista ja muualta.

Kirjoitus on alunperin ollut osa englan-

[8] http://www.cs.helsinki.fi/teemu.roos/pub/brazil.pdf

ninkielistä luentomateriaalia, joka on kokonaisuudessaan saatavilla kirjoittajan kotisivulta[8]. Kirjoittaja on hyötynyt määrättömästi etenkin Jorma Rissasen ja Peter Grünwaldin kanssa käymistään MDL-periaatteeseen liittyvistä keskusteluista (ja väittelyistä). Kiitokset myös Tietojenkäsittelytiede-lehden toimitukselle ja etenkin Antti Valmarille merkittävästä panoksesta, joka on auttanut parantamaan kirjoituksen muotoa. Tässä kirjoituksessa esitetyt näkemykset ovat tietysti omalla vastuullani, eivätkä ne kaikilta osin perustu tosielämän tapahtumiin. Ensimmäisen sivun neljä symbolia ovat pystyakselin suunnassa peilattuja numeroita esittäviä piirroksia. Seuraava symboli näyttää jotakuinkin tältä: ಠ.

Viitteet

1. Thomas M. Cover ja Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, New York, USA, 1991.
2. Peter D. Grünwald. *The Minimum Description Length Principle*. MIT Press, 2007.
3. Petri Kontkanen ja Petri Myllymäki. A linear-time algorithm for computing the multinomial stochastic complexity. *Information Processing Letters* 103(6):227–233, 2007.
4. Leon G. Kraft. *A Device for Quantizing, Grouping, and Coding Amplitude-Modulated Pulses*. Master's thesis, Massachusetts Institute of Technology, Cambridge, USA, 1949.
5. Ming Li ja Paul M. B. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer-Verlag, Berliini, Saksa, 1993.
6. Brockway McMillan. Two inequalities implied by unique decipherability. *IRE Transactions on Information Theory*, 2(4): 115–116, 1956.
7. Karl Popper. *The Logic of Scientific Discovery*. Hutchinson & Co., Lontoo, Iso-Britannia, 1. englanninkielinen painos, 1959.
8. Jorma Rissanen. Generalized Kraft inequality and arithmetic coding. *IBM Journal of Research and Development*, 20 (3):198–203, 1976.
9. Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.
10. Jorma Rissanen. A universal prior for integers and estimation by minimum description length principle. *Annals of Statistics*, 11(2):416–431, 1983.
11. Jorma Rissanen. *Stochastic Complexity in Statistical Inquiry*. World Scientific Publishing Company, New Jersey, USA, 1989.
12. Jorma Rissanen. Information theory and neural nets. Smolensky, Mozer ja Rumelhart (toim.). *Mathematical Perspectives on Neural Networks*. Lawrence Erlbaum Associates, Mahwah, USA, 1996.
13. Claude E. Shannon. A Mathematical Theory of Communication. *Bell Systems Technical Journal*, 27:379–423, 623–656, 1948.
14. David Solomon. *Data Compression: The Complete Reference*. Springer, New York, USA, 3. painos, 2004.
15. Chris S. Wallace ja David M. Boulton. An information measure for classification. *Computer Journal*, 11(2):185–194, 1968.
16. Arnold Zellner, Hugo A. Keuzenkamp ja Michael McAleer (toim.). *Simplicity, Inference and Modelling: Keeping it Sophisticatedly Simple*. Cambridge University Press, Cambridge, Iso-Britannia, 2001.