MATA2500 – Informaatioteoria – Syksy 2019

ℝami Luisto

3. maaliskuuta 2020

Sisältö

Tämä on työn alla oleva kurssimateriaali Syksyn 2019 kurssille MATA2500 – Informaatioteoria. Kurssin on jaettu kahteen osaan jotka ovat perusosa sekä matemaattinen laajennus, kumpikin kahden opintopisteen laajuisia. Matemaattinen laajennus syventää perusosan huomioita, ja siihen liittyvät aiheet ovat omissa aliluvuissaan.

Caveat lector 1:

Luennoitsija on topologi eikä ole tietojenkäsittelytieteen, fysiikan tai tilastotieteen erityisosaaja. Luennoilla ja materiaalissa olevat väitteet tulevat olemaan vahvalla matemaattisella perustalla, mutta yhteydet muihin aloihin saattavat olla vanhentuneella tietopohjalla.

Caveat lector 2:

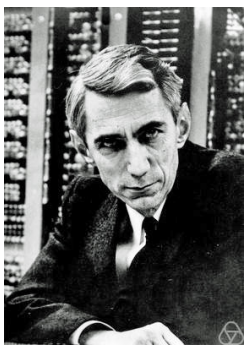
Tätä monistetta pyritään tahkoamaan ulos samaan tahtiin kuin mitä materiaalia käydään jotta luennon missanneetkin voivat asiaan perehtyä. Tästä seuraa että näitä juntataan aikamoista haipakkaa, eli tyyli saattaa ajoittain olla hieman puhekielistä. Pahoittelen tyylipuutteita jotka toivottavasti korjataan viimeistään joululomalla – älä käytä tämänhetkistä versiota mallina hyvästä kirjoitusasusta kandissasi/vastaavassa.

Luku 1

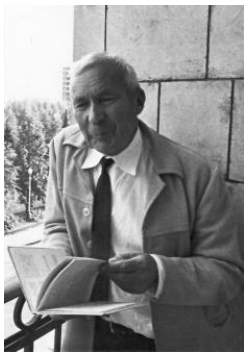
Johdattelua informaation käsitteeseen

Kurssin tarkoituksena on käsitellä informaatiota matemaattisena käsitteenä. Kurssilla ei kuitenkaan tulla antamaan suoraa määritelmää "Informaatio on ..."vaan informaatiota lähestytään useamman eri käsitteen kautta. Tärkeimmät näistä tulevat olemaan Shannonin entropia sekä Kolmogorov-kompleksisuus joista annetaan tässä luvussa hieman esimakua. Informaatioteorialla on suoria sovelluksia niin tietoliikenteeseen, kryptografiaan kuin koneoppimiseenkin. Näihin sovelluksiin tullaan palaamaan kurssin edetessä.

Kuvien lähde: Wikimedia commons



Kuva 1.1: Claude Shannon



Kuva 1.2: Andrej Kolmogorov

1.1 Perusosa

Sanaa *informaatio* käytetään tällä kurssilla sekä formaaleissa että vähemmän formaaleissa yhteyksissä. Tässä kappaleessa sana informaatio ei tarkoita vielä mitään matemaattisesti formaalia asiaa vaan tavallista puhekielen ilmaisua.

1.1.1 Informaatio epävarmuuden vähentymisenä

Ensimmäinen lähestymissuuntamme kohti informaatiota perustuu ideaan, että informaation saaminen vastaa epävarmuuden vähentymistä. Mikäli et muista serkkusi syntymäpäivää, mutta setäsi mainitsee että se on tammikuussa, on setäsi välittänyt sille informaatiota sillä epävarmuutesi vähenee. Jos setäsi toistaa asian hetken päästä uudestaan, ei informaatiota enää välity koska epävarmuutesi määrä pysyy samana. Otetaan konkreettinen esimerkki.

Esimerkki 1. Pöydälle on asetettuna 16 korttia jotka on numeroitu 0–15. Kortit 0-5 ovat vihreitä, kortit 6-7 punaisia ja loput sinisiä. Täsmälleen yhden kortin taakse on piirretty hymiö. Mikä seuraavista väitteistä antaa eniten tietoa hymiön sijainnista?

(K1) Hymiö ei ole sinisen kortin takapuolella.

(K2) Hymiö on punaisen kortin takapuolella.

(K3) Hymiö on kortin 6 takapuolella.

Entä jos ennen testin alkua kaverisi olisi käynyt kurkkaamassa kortin 15 takapuolta ja kertonut sinulle että hymiö ei ole siellä, välittävätkö ylläolevat viestit tällä lisätiedolla nyt enemmän tai vähemmän informaatiota? Entä jos olet istunut monilla luennoilla ja huomannut että tällaiset esimerkit on usein rakennettu tarkoitushakuisesti, ja olisit valmis lyömään kaverisi kanssa kympistä vetoa että hymiö on jommankumman punaisista korteista takana?

Tämä lähestymistapa, jossa arvioimme viestin informaatiosisältöä sen perusteella paljonko epävarmuutta viesti poistaa, johdattaa meidät Shannonin entropian pariin. Luontainen konteksti tälle käsittelylle on todennäköisyysteoria, jossa todennäköisyyksillä ei kuvata satunnaisia tapahtumia vaan epävarmuuden määrää. Vastauksena esimerkin kysymykseen, taulukossa **??** on kerrottuna montako Shannon-bittiä ylläolevassa tilanteessa välittyy.

1.1.2 Informaatio kuvailun hankaluutena

Olet matkailemassa Lapin tuntureilla, jossa puhelinyhteydet ovat heikot. Sinulla on mukana vain vanha Nokian 3310 jolla voit, vaivalloisesti, lähettää tekstiviestejä. Kesken matkan kämppäkaverisi kysyy sinulta viestillä että mikä olikaan yhteisen Netflixinne salasana. Mikä seuraavista viesteistä on helpompi välittää?

	Taustaoletukset		
	Ei esitietoja	Ei kortin 15 takana	50-50 punaisen takana
(K1)	1 bit	≈ 0.9 bit	≈ 0.5 bit
(K2)	3 bit	≈ 2.9 bit	1 bit
(K3)	4 bit	≈ 3.9 bit	2 bit

Taulukko 1.1: Esimerkissä 1 välitetyn informaation määärä Shannon-biteissä. Opitaan laskemaan myöhemmin.

(V1) "Salasana on `AbcAbcAbcAbcAbcAbcAbcAbcAbcAbcAbcAbcAbc`."

(V2) "Salasana on `3.14159265358979323846264338327950288841`."

(V3) "Salasana on `ZLbXCvQqsvEeGFsebVjDHgefaqUSXPfTkAcDJVn`."

Näistä viesteistä huomataan että vaikka kaikkien pituus on tasan 57 merkkiä, niin kaksi ensimmäistä voidaan viestiä lyhyemmin:

1. "Salasana on 'Abc' 13 kertaa peräkkäin." (36 merkkiä)
2. "Salasana on piin arvo 38 desimaalin tarkkuudella." (49 merkkiä)

Kolmas viesti kuitenkin vaikuttaa siltä, että sen lyhentäminen on vähintään epätriviaalia ellei suorastaan mahdotonta, ainakin varsinaisen salasanan osalta. Kysymys siitä, että voiko viestin sisällön välittää lyhyemmässä muodossa kytkeytyy Kolmogorov-kompleksisuuteen, joka on toinen luonnollinen informaation mitta.

Kolmogorov-kompleksisuuden määritelmässä on hyvin kriittistä tietää sallitut kuvailukeinot. Esimerkiksi jos kämppäkaveri osaa Python-kielen perusteet, voidaan ensimmäistä viestiä tiivistää edelleen muotoon

`"Salasana on print(13*"Abc")."`(28 merkkiä),

mutta jos hän ei tiedä mitään ohjelmoinnista ei kyseinen viesti välttämättä auta. Viestin (V2) yksinkertaiseen lähettämiseen puolestaan tarvitaan se, että sekä lähettäjä että vastaanottaja tietävät (tai osaavat laskea) piin desimaaleja tarvittavalla tarkkuudella. Tiettyä tarkkuutta vaaditaan myös 'hyvinmääriteltävyyden' takaamiseksi; mitä numeroa kuvaa ilmaisu "Salasana on pienin numero jota ei voi kuvailla alle sadalla sanalla."?

1.1.3 Informaatioteorian sovelluskohteista

Informaatioteoria on kiinnostavaa itsessään, mutta sillä on myös useita sovelluskohteita. Mainitsemme tässä lyhyesti muutamia ja palaamme niihin kurssin edetessä syvemmin.

Tietoliikenne

Informaatioteorian perustajana pidetään Claude Shannonia, joka kirjoitti nykypäivänäkin lukemisen arvoisen paperin [Sha48] työskennellessää Bell Labsissä. Shannonin tulokset antoivat ensimmäisiä rajoja viestiliikenteen maksimikapasiteetille. Myös pakkausalgoritmeille saadaan käytännön rajoja, mikä tuo yhden liitoskohdan Shannonin entropian sekä Kolmogorov-kompleksisuuden välille.

Kolmogorov-kompleksisuus esiintyy myös luontevasti (ja formaalimmalla pohjalla) teoreettisemman tietojenkäsittelytieteen puolella; suosittelen kovasti kurssia *TIEA241 Automaatit ja kieliopit* jos haluaa saada tiukan otteen Kolmogorov-kompleksisuuteen ja Turing-täydellisyyteen.

Kryptografia

Kryptografian keskiössä on muodostaa menetelmiä joiden avulla kaksi tahoa voivat viestiä keskenään turvallisesti, eli siten että viestintäkanavaa mahdollisesti kuuntelevan tahon on erityisen vaikeaa saada selville viestien sisältöä.

Salausalgoritmit tarjoavat tähän yhden vaihtoehdon. Tarkoituksena on muodostaa algoritmit S ja P siten että annettu viesti v voidaan salata algoritmilla S käyttäen salausavainta $\mathbf{a}$ ja salattu viesti $S(v, \mathbf{a})$ purkaa algoritmilla P takaisin alkuperäiseksi viestiksi salausavaimen $\mathbf{p}$ avulla. Yksi tapa yrittää tehdä viestinvaihdosta tällaisen salausmenetelmän kautta on turvallista, tulee salatun viestin $S(v, \mathbf{a})$ välittää mahdollisimman vähän informaatiota viestistä v tilanteessa, jossa salausavainta $\mathbf{p}$ ei tunneta.

Informaatioteorian avulla voidaan sekä analysoida salausmenetelmien luotettavuutta että todistaa joitain rajoitteita niiden toimivuudelle.

Tilastollinen päättely

Kun tieteilijän edessä on hypoteesi, hän suorittaa kokeen. Mutta minkä kokeen? Jotkin koejärjestelyt saattavat olla helppoja ja nopeita suorittaa, mutta usein kokeen suorittaminen voi olla hidasta, kallista tai ainutkertainen mahdollisuus. Tällöin on taloudellista kuluttaa resursseja kokeen suunnitteluun ja valintaan. Hyvä koesuunnittelu on oma alansa, kts. esim. *TILS655 Koesuunnittelu*, mutta esimerkiksi nk. D-Optimointi nojaa Shannonin (differentiaali)entropian maksimointiin.

Myöskin tilastollisia tuloksia analysoidessa Bayesin kaavan käyttö tulee käymään tutuksi, mutta miten määrittää priorijakauma ennen ensimmäistäkään datapistettä? Eräs suosittu ratkaisu on käyttää maksimientropian periaatetta, joka jälleen kytkeytyy Shannonin informaatiokäsitteisiin.

1.1.4 Muutama sana todennäköisyysteorian tulkinnasta

Kurssilla tullaan käyttämään todennäköisyyslaskentaa useammassa eri yhteydessä. Tekniseltä kannalta emme tule tarvitsemaan juuri lukiokursseja laajempaa osaamista, mutta käytämme todennäköisyyslaskentaa hieman eri *tulkinnalla* kuin mitä yleensä lukiossa. Laskut tulevat pysymään tismalleen saman näköisinä, mutta toinen lähestymistapa saattaa olla osalle avartava. Jos taas uusi tulkinta ahdistaa, niin älä huoli, formaalisti kaikki tehdään aivan samalla tavalla tulkinnasta riippumatta.

Uhkapelejä ja frekvenssejä

Todennäköisyyslaskennan historia lähtee uhkapeleista – noppien ja kolikoiden heitoista ja vastaavista – ja todennäköisyyslaskenta soveltuukin loistavasti tällaisten tilanteiden analysointiin. Se ei kuitenkaan tarkoita että tämä on ainut viitekehys jossa todennäköisyyslaskentaa voi käyttää. Erityisesti tämän kurssin kannalta meidän on tärkeää keskittyä ajatukseen, että todennäköisyyslaskentaa voi soveltaa myös tilanteissa joissa ei ole mitään satunnaista.

Verrataan seuraavia tapahtumia:

1. Luennoitsija aikoo heittää kohta tavallista noppaa. Mikä on todennäköisyys, että saadaan luku 6?
2. Luennoitsija heitti ennen luennon alkua tavallista noppaa, mutta ei suostu kertomaan tulosta. Mikä on todennäköisyys, että saatu luku oli 6?
3. Luennoitsija kertoo että hänellä on laukussaan 1-6 nallekarkkia, muttei suostu kertomaan tarkkaa määrää. Mikä on todennäköisyys että hänellä on niitä tasan 6?
4. Edessäsi on kuusi numeroitua ovea joista tasan yhden takana on vuohi. Mikä on todennäköisyys sille, että vuohi on oven 6 takana?

Jokaisen tapahtuman kohdalla tuntuisi jokseenkin luontevalta sanoa että todennäköisyys on $\frac{1}{6}$, vaikka vain ensimmäiset kaksi tilannetta pohjautuvat suoraan satunnaisuuteen. (Ja näistäkin jälkimmäisessä "satunnaisuus on jo tapahtunut".) Yksi tapa yrittää tulkata esirerkiksi tapahtumaa kaksi aiemmalle kielelle on miettiä frekvenssejä eli kuvitella että mikäli nopanheitto ja sen jälkeen arvuuttelu toistettaisiin useita kertoja, niin mikä olisi silmälukujen 6 keskimääräinen osuus. Tämä frekvenssilähestyminen toimii, mutta joissain tilanteissa se voi tuntua hieman teennäiseltä:

> *Innostuneena uudesta kaukoputkestasi mittaat usean vuoden ajan Saturnuksen paikkaa taivaalla. Kerättyäsi roimasti datapisteitä ja luettuasi kiertoratamekaniikkaa tulet tulokseen että Saturnuksen massan on oltava* $5 \pm 1 \times 10^{26} kg$.

Tässä tilanteessa mittauksen tarkkuuteen liittyvän todennäköisyyden tulkinta frekvenssinä vaikuttaa jo hieman kömpelöltä; Saturnuksen massa voidaan toki tulkita satunnaislukuna joka vaihtelee vaikkapa kuvitteellisten rinnakkaisuniversumien välillä, mutta täysin tyydyttävältä tämä tulkinta ei vaikuta.

Bayesiläistä päättelyä

Tällä kurssilla lähestytäänkin todennäköisyyttä ajatuksella, että se kertoo miten paljon kuvittelemme jonkin asian olevan totta. (Degrees of Belief.) Toisin ajateltuna todennäköisyydet kuvaavat epävarmuuttamme jonkin asian suhteen. Mitä suurempi todennäköisyys, sitä varmempia olemme asian varsinaisesta tilasta.

Katsotaan aiempia kysymyksiä uudessa valossa:

1. Edessäsi on kuusi numeroitua ovea joista tasan yhden takana on vuohi. Miten varma[1] olet siitä, että vuohi on oven 6 takana?

2. Luennoitsija kertoo että hänellä on laukussaan 1-6 nallekarkkia, muttei suostu kertomaan tarkkaa määrää. Miten luultavasti on niitä tasan 6?

3. Luennoitsija heitti ennen luennon alkua tavallista noppaa, mutta ei suostu kertomaan tulosta. Kuinka vahvasti luulet että saatu luku oli 6?

4. Luennoitsija aikoo heittää kohta tavallista noppaa. Miten varma olet siitä, että saadaan luku 6?

Tällaista lähestymistapaa, jossa todennäköisyyttä pidetään 'varmuuden määränä' (Degrees of Belief), kutsutaan usein todennäköisyyden Bayesiläiseksi tulkinnaksi. Bayesiläinen lähestymistapa voi vaikuttaa alkuun hieman epämatemaattiselta, sillä 'varmuuden määrä' on kovin subjektiivinen käsite. Syventävässä osiossa keskusteltava Coxin lause kuitenkin takaa, että tämän 'varmuuden määrän' voi ajateella olevan se varmuuden määrä mitä täydellinen päättelijä tilanteessa tuntisi. (Matemaattisen ultimaattinen Sherlock Holmes.)

Myöskin, kolikon kääntöpuolena huomaamme että Degrees of Belief -lähestymistapa voi tuntua hieman kömpelöltä silloin kun tutkitaan satunnaista tilannetta kuten nopan heittoa. Mikään ei kuitenkaan estä sinua tulkitsemasta todennäköisyyksiä eri kontekstissa (tai eri tyyppisinä todennäköisyyksinä) tilanteesta riippuen.

[1] Tilanteessa, jossa todennäköisyydet ovat reippaasti alle puolen, voivat termit "varma" taikka "luultava" tuntua hieman vierailta, mutta emme rupea tässä keksimään uusia sanoja vaan yritämme kestää.

1.1.5 Todennäköisyyslaskennan käsitteiden pikakertaus

Kirjataan ylös muutamia peruskäsitteitä ja huomioita mitä tulemme todennäköisyyslaskennasta kurssilla tarvitsemaan. Käsittelemme kurssilla käytännössä pelkästään diskreettejä taikka äärellisiä satunnaismuuttujia, joten muotoilemme perustermistön niiden asetelmassa.

Määritelmä 1.1.1. (Äärellinen) todennäköisyysavaruus (tai todennäköisyysjakauma) on kolmikko (Ω, F, P) missä Ω on jokin epätyhjä äärellinen joukko, $F \subset \mathcal{P}(\Omega)$ epätyhjä kokoelma joukon Ω osajoukkoja joka sisältää alkioidensa komplementit, yhdisteet sekä leikkaukset ja $P\colon F \to \mathbb{R}$ funktio jolle pätevät seuraavat ominaisuudet.

(K1) $P(A) \in [0,1]$ kaikilla $A \in F$.

(K2) $P(\Omega) = 1$ ja $P(\emptyset) = 0$.

(K3) Jos $A \cap B = \emptyset$, niin $P(A \cup B) = P(A) + P(B)$.

Tällä kurssilla usein, muttei aina, $F = \mathcal{P}(\Omega)$. Joukon Ω alkioita kutsutaan *alkeistapahtumiksi* ja sen osajoukkoja kokoelmassa F *tapahtumiksi*. Merkitsemme ajoittain $P(\{x\}) = P(x)$ ja kutsumme lukua $P(A)$ tapahtuman A todennäköisyydeksi.

Huomaa, että diskreetissä todennäköisyysjakaumassa kuvaus P määräytyy täysin sen arvoista alkeistapahtumissa $P(x)$, $x \in \Omega$. Tällä kurssilla alkeistapahtumien joukko Ω on usein luonnollisella tavalla indeksöity, esimerkiksi $\Omega = \{a_1, \ldots, a_k\}$, jolloin kirjoitamme $P(a_j) = p_j$.

Esimerkki 2. Luennoitsijan laukussa olevien nallekarkkien lukumäärään liittyvää epävarmuutta voidaan kuvata todennäköisyysjakaumalla

$$(\{\text{yksi nallekarkki}, \ldots, \text{kuusi nallekarkkia}\}, \mathcal{P}(\Omega), P),$$

missä P on vakiokuvaus $\frac{1}{6}$.

Tilanteessa jossa joku on saanut selville, että luennoitsija on nallekarkkifani, saattaisi tilannetta mallintaa paremmin kolmikko missä todennöisyydet ovat $(1/32, 1/32, 1/16, 1/8, 1/4, 1/2)$.

Määritelmä 1.1.2. *Yhteisjakauma* on todennäköisyysjakauma muotoa

$$(\Omega_1 \times \Omega_2, F_1 \times F_2, P).$$

Mikäli $A \in F_1$ ja $B \in F_2$, niin merkitsemme $P(A \times B) =: P(A, B)$ ja sanomme että tämä on todennäköisyys sille että sekä A että B tapahtuvat.

Yhteisjakaumassa tapahtuman $A \in F_1$ *marginaalinen todennäköisyys* on

$$P(A) = \sum_{B \in F_2} P(A, B).$$

Huomaa että marginaaliset todennäköisyydet tuottavat uudet todennäköisyysjakaumat (Ω_i, F_i, P_i), $i = 1, 2$.

Edelleen yhteisjakaumassa tapahtuman A ehdollinen todennäköisyys tapahtuman B suhteen on

$$P(A|B) = \frac{P(A,B)}{P(B)}.$$

Huomaa, että yhteisjakaumassa pätee

$$P(A,B) = P(A|B)P(B).$$

Esimerkki 3. Luennoitsijalla on edessään 6 numeroitua rasiaa, joista 1-2 ovat isoja ja loput 4 pieniä. Hän kertoo valinneensa umpimähkään huoneestaan joko jalkapallon tai golfpallon ja piilottaneensa sen johonkin rasiaan. Huomaat, ettei jalkapallo voi mahtua pienempiin rasioihin ja mallinnat tilannetta yheisjakaumalla, jossa $\Omega_1 = \{\text{Jalkapallo}, \text{Golfpallo}\}$, $\Omega_2 = \{1, \ldots, 6\}$ ja todennäköisyys P on

$$P(\text{Jalkapallo}, k) = \begin{cases} \frac{1}{4}, & \text{kun } k \in \{1,2\} \\ 0, & \text{kun } k \geq 3. \end{cases}, \quad P(\text{Golfpallo}, k) = \frac{1}{12}.$$

Merkataan tapahtumaa "luennoitsija valitsi golfpallon" A ja tapahtumaa "valittiin rasia 3" B. Nyt voidaan laskea marginaaliset todennäköisyydet

$$P(A) = \sum_{k \in \Omega_2} P(A,k) = 6\frac{1}{12} = \frac{1}{2}$$

ja

$$P(B) = \sum_{x \in \Omega_1} P(x,B) = \frac{1}{4} + 0 = \frac{1}{4}.$$

Edelleen voidaan laskea ehdolliset todennäköisyydet

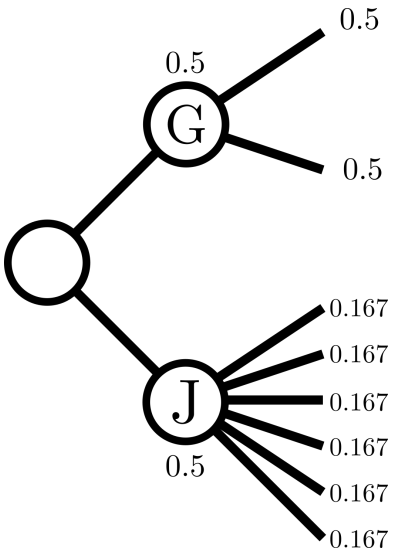
$$P(A|B) = \frac{P(A,B)}{P(B)} = \frac{\frac{1}{12}}{\frac{1}{12}} = 1$$

ja

$$P(B|A) = \frac{P(A,B)}{P(A)} = \frac{\frac{1}{12}}{\frac{1}{2}} = \frac{1}{6}.$$

Tärkeä tulos päättelyä varten on seuraava Bayesin kaava:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}.$$



Kuva 1.3: Esimerkin 3 todennäköisyyspuu

Jatkossa suosimme ilmaisua $P(X = a_j|I)$ ilmaisun $P(X = a_j|I)$ sijaan. Tällä korostamme ja pidämme mielessä että lähestymistapamme on degrees of belief -hengessä jolloin taustalla on aina jotain informaatiota I. Täten esimerkiksi Bayesin kaava on hyvä kirjoittaa muodossa

$$P(A|B, I) = \frac{P(B|A, I)P(A|I)}{P(B|I)}.$$

Määritelmä 1.1.3. Olkoon (Ω, F, P) todennäköisyysavaruus.
Satunnaismuuttuja on kuvaus $X \colon \Omega \to \mathbb{R}$. Merkitsemme

$$P(X \in B) := P(\{a \in \Omega \mid X(a) \in B\}), \quad P(X = y) := P(\{a \in \Omega \mid X(a) = y\}).$$

Satunnaismuuttujalle X määritellään *odotusarvo* $E(X)$ asettamalla

$$E(X) = \sum_{a \in \Omega} P(X = a)X(a).$$

1.1.6 Entropia ja informaatiosisältö, lyhyesti

Tulemme käyttämään kurssilla paljon aikaa perustellaksemme miksi seuraava määritelmä on hyvä idea.

Määritelmä 1.1.4. Olkoon (Ω, F, P) äärellinen todennäköisyysjakauma. Tällöin tapahtuman $A \in F$, missä $P(A) > 0$, *informaatiosisältö* on

$$h(A) := \log_2(\frac{1}{P(A)})$$

Alkeistapahtumille merkitsemme jälleen $h(\{a_i\}) = h(a_i)$, tai vielä lyhyemmin $\log_2(1/p_i)$. Informaatiosisällön yksikkö on meille *(Shannonin) bitti*.

Todennäköisyysjakauman *entropia* on satunnaismuuttujan h odotusarvo, eli

$$H := E[h] = -\sum_i p_i \log_2(p_i).$$

Entropiaa laskettaessa, mikäli tapahtuman todennäköisyys on 0, tulkitaan että $0 \times \log_2(0) = 0$. Entropian yksikkö on myöskin Shannonin bitti.

Muista, että $-\log_2(x) = \log_2(1/x)$. Huomaa myös, että logaritmin kannan valinta vaikuttaa vain globaalilla vakiokertoimella,

$$\log_2(x) = \log_{10}(x)/\log_{10}(2) = \ln(x)/\ln(2),$$

ja on luonteeltaan kuin mittayksikön valinta. Käytämme pertinteikkäässä hengessä kantalukua 2.

Lasketaan informaatiosisältöä ja entropiaa aiemmin ilmenneiden esimerkkien tilanteissa.

Esimerkki 4.

1. Kuusitoista korttia joista yhden takana on hymiö. Kukin kortti oli, a priori, yhtä todennäköinen joten yksittäisen kortin paljastuminen tasaisesta jakaumasta antaa informaatiosisällön $-\log_2(1/16) = \log_2(16) = 4$. (Huomaa, että olisit saanut kortin paikan selville neljällä kyllä/ei kysymyksellä – kts. binäärihaku.)

   Sen paljastaminen, että hymiö ei ollut sinisen kortin takana sulki pois 8 vaihtoehtoa 16:a, ja koska hymiön paikasta ei a priori tiedetty mitään, oli tapahtuman todennäköisyys $8/16 = 0.5$. Tämän tapahtuman informaatiosisältö oli siis $\log_2(2) = 1$. Kun suljetaan pois puolet yhtä epävarmoista vaihtoehdoista, saadaan yksi bitti informaatiota.

   Todennäköisyysjakauman entropiaksi saadaan vastaavasti

$$H = -\sum_{i=0}^{15} \frac{1}{16} h(\frac{1}{16}) = \sum_{i=0}^{15} \frac{1}{16} \log_2 16 = 16\frac{1}{16}4 = 4.$$

2. Kuusi ovea, joista yhden takana on vuohi. Vuohen paljastaminen tuottaa $-\log_2(1/6) = \log_2(6) \approx 2.6$ bittiä informaatiota. Informaatiosisällön ei tarvitse olla kokonaisluku. Jakauman entropia on

$$H = -\sum_{i=0}^{5} \frac{1}{6} h(\frac{1}{6}) = h(6) \approx 2.6.$$

3. Esimerkki 3; golfpallo ja jalkapallo.

Yhteisjakauman entropiaksi saadaan

$$\begin{aligned} H &= \sum_{x\in\Omega_1, k\in\Omega_2} P(x,k)h(P(x,k)) \\ &= \sum_{x\in\Omega_1}\sum_{k=1}^{6} P(x,k)h(P(x,k)) \\ &= \sum_{k=1}^{6} P(\mathrm{G},k)h(P(\mathrm{G},k)) + \sum_{k=1}^{6} P(\mathrm{J},k)h(P(\mathrm{J},k)) \\ &= \sum_{k=1}^{6} \frac{1}{12}\log_2(12) + \left(\frac{1}{4}\log_2(4) + \frac{1}{4}\log_2(4) + 0 + \ldots + 0\right) \\ &= \frac{1}{2}\log_2(12) + \frac{1}{2}\log_2(4) \approx 2.8. \end{aligned}$$

Tutkikaan seuraavaksi monimutkaisempaa esimerkkiä Shannonin informaation käytöstä. Tässä kohtaa ei ole vielä selvää että miksi kukin vaihe suoritetaan, tarkoituksena on antaa seurattavavissa oleva esimerkki siitä, miten informaatiosisällön käsitettä voi käyttää.

Esimerkki 5. Sinulle annetaan 12 palloa ja vaaka, jolla voi vertailla kahdessa vaakakupissa olevien asioiden painoja. (Saat siis laittaa vaan kahteen kuppiin haluamasi määrän palloja, ja vaaka kertoo ovatko kuppien sisällöt samanpainoisia, ja jos eivät niin kumpi on painavampi.) Sinulle kerrotaan että 11 palloista ovat tismalleen samanpainoisia, ja yksi joko kevyempi tai painavampi kuin muut. Miten monta punnitusta tarvitset selvittääksesi mikä palloista on poikkeava ja että onko se painavampi vai kevyempi kuin muut?

Tutkitaan ongelmaa entropian avulla. Määritellään alkuun tilannetta kuvaama todennäköisyysjakauma. Numeroidaan pallot 1–12 ja merkitään mahdollisuutta "pallo k on painavampi kuin muut" merkillä $k+$ ja mahdollisuutta "pallo k on kevyempi kuin muut" merkillä $k-$. Nyt tilannetta kuvaava todennäköisyysjakauma on (Ω, F, P), missä

$$\Omega = \{1-, 2-, \ldots, 12-, 1+, 2+, \ldots, 12+\}, \quad F = \mathcal{P}(\Omega)$$

ja $P(x) = \frac{1}{24}$ kaikilla $x \in \Omega$.

Vastauksen selvittämiseksi sinun täytyy pystyä sanomaan "pallo k on kevyempi/painavampi kuin muut"; tämän tapahtuman informaatiosisältö on $h(\frac{1}{24}) = \log_2(24) \approx 4.58$. Meidän täytyy siis saada ulos ainakin (täsmälleen) tämän verran informaatiota ratkaistaksemme ongelman. Mutta miten monta bittiä informaatiota yksi punnitus antaa? Mikä punnitus meidän kannattaa aluksi suorittaa?

6×6 Emme tiedä mitään pallojen painoista aluksi, joten voimme olettaa että vertaamme vaa'alla palloja $1-6$ (vasen kuppi) palloihin $7-12$ (oikea

kuppi). Meillä on kaksi vaihtoehtoa; joko vasen kuppi on painavampi tai oikea kuppi on painavampi. (Tasapainotila ei ole mahdollinen, eli sen todennäköisyys on nolla.)

Vasen kuppi on painavampi vastaa tapahtumaa $\{1+, \ldots, 6+, 7-, \ldots, 12-\}$, ja sen todennäköisyys on $12/24 = 0.5$. Vastaavasti oikea kuppi painavampi vastaa tapahtumaa $\{1-, \ldots, 6-, 7+, \ldots, 12+\}$, jonka todennäköisyys on myös 0.5. Tätä punnitusta voidaan siis ajatella omana todennäköisyysjakaumanaan, jossa on kolme alkiota "vasen painavampi, tasapainossa, oikea painavampi" ja näitä vastaavat todennäköisyydet 0.5, 0 sekä 0.5. Tämän satunnaisjakauman entropiaksi saadaankin siis

$$H = -0.5 \log_2(0.5) + 0 - 0.5 \log_2(0.5) = 1.$$

Tulkintamme siis on, että 6 vastaan 6 -mittauksen voidaan olettaa antavan noin yhden bitin verran informaatiota.

5×5 Emme tiedä mitään pallojen painoista aluksi, joten voimme olettaa että vertaamme vaa'alla palloja $1-5$ (vasen kuppi) palloihin $6-10$ (oikea kuppi). Meillä on kolme vaihtoehtoa; joko vasen kuppi on painavampi, oikea kuppi on painavampi, tai kupit ovat tasapainossa.

Vasen kuppi on painavampi vastaa tapahtumaa $\{1+, \ldots, 5+, 6-, \ldots, 10-\}$, ja sen todennäköisyys on $10/24 = 5/12$. Vastaavasti oikea kuppi painavampi vastaa tapahtumaa $\{1-, \ldots, 5-, 6+, \ldots, 10+\}$, jonka todennäköisyys on myös $5/12$. Tasapainotila saavutetaan tapahtumassa $\{11-, 12-, 11+, 12+\}$, ja tämän tapahtuman todennäköisyys on $4/24 = 1/6$.

Tätäkin punnitusta voidaan ajatella omana todennäköisyysjakaumanaan, jossa on kolme alkiota "vasen painavampi, tasapainossa, oikea painavampi" ja näitä vastaavat todennäköisyydet $5/12$, $1/6$ sekä $5/12$. Tämän satunnaisjakauman entropiaksi saadaankin siis

$$H = \frac{5}{12} \log_2(\frac{12}{5}) + \frac{1}{6} \log_2(6) + \frac{5}{12} \log_2(\frac{12}{5}) \approx 1.48.$$

Mittaus 5 vastaan 5 antaa siis keskimäärin enemmän informaatiota kuin 6 vastaan 6 -mittaus.

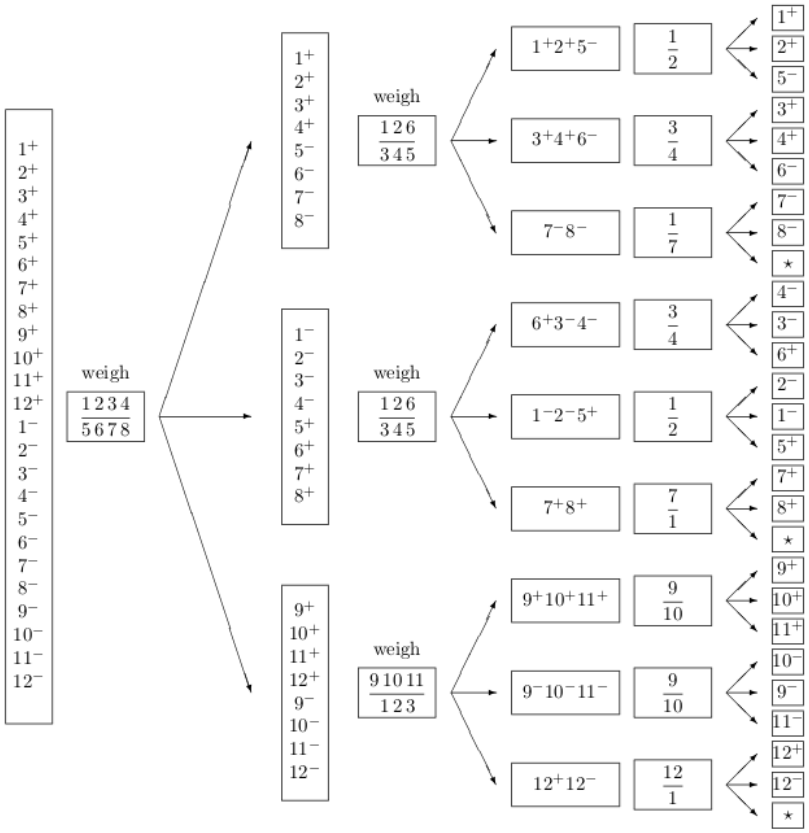
4×4 Kuten edellä, tässä mittauksessa päädytään todennäköisyysjakaumaan jossa on kolme alkiota "vasen painavampi, tasapainossa, oikea painavampi" ja näitä vastaavat todennäköisyydet $1/3$, $1/3$ sekä $1/3$. Entropia on tällöin

$$H = 3(\frac{1}{3} \log_2(3)) = \log_2(3) \approx 1.58.$$

Loput valinnat Emme toista samaa laskua, mutta loput mittausvaihtoehdot 3×3, 2×2 ja 1×1 antavat kaikki vähemmän informaatiota kuin 4 vastaan 4-mittaus.

Laskujen perusteella huomaamme, että 4 vastaan 4 -mittaus antaa meille suurimman informaatiosisällön odotusarvon. Tulemme myöhemmin huomaamaan, että äärellisissä todennäköisyysjakaumissa entropia on sitä suurempi mitä lähempänä se on tasajakaumaa, joka maksimoi entropian. Erityisesti emme voi toivoa punnitukselta, jossa on kolme vaihtoehtoa, suurempaa entropiaa eli keskimääräistä informaatiosisältöä ei voi saavuttaa. Tästä seuraa, että koska yhden punnituksen maksimientropia on ≈ 1.58, niin punnituksia tarvitaan vähintään $4.58/1.58 \approx 2.9$ eli vähintään kolme kappaletta. (Tämä ei takaa, että kolme mittausta aina riittäisi, mutta vähempi ei ainakaan ole tarpeeksi.)

Tultuamme tulokseen että kannattaa aloittaa 4 vastaan 4 -mittauksella, niin mitä seuraavaksi? Vastaus on, että katsomme kutakin mahdollista punnitustulosta "vasen painavampi, tasapainossa, oikea painavampi" erikseen, ja teemme kussakin tapauksessa uuden analyysin että mikä punnitustapa antaisi suurimman entropian, eli keskimäärin eniten informaatiota. Ratkaisua kannattaa pohtia ihan itse, mutta kolme punnitusta onnistuu esimerkiksi kuten kuvassa 1.4.



Kuva 1.4: Punnitusongelman optimaalinen ratkaisu. Kuva D. MacKayn kirjasta [Mac03].

1.2 Syventävä laajennos

Tässä osiossa perehdytään Coxin lauseeseen jonka nojalla meidän kannattaa jatkossakin nojata todennäköisyysteoriaan tehdessämme päättelyitä. Tämän jälkeen tutkimme hieman muutamaa Monty Hall -henkistä ongelmaa ja katsomme mitä informaatiolle näissä esimerkeissä käy.

1.2.1 Coxin lause

Aloitamme muutamalla lainauksella.

> 'La théorie des probabilités n'est que le bon sens reduit au calcul.'
> -Laplace (1814)

(Vapaasti suomennettuna; Todennäköisyyslaskenta on vain käytännön järkeä esitettynä laskuilla.)

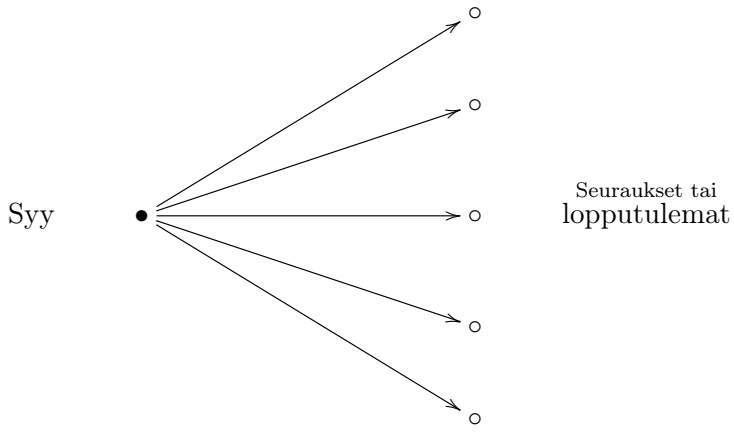
> The actual science of logic is conversant at present only with things either certain, impossible, or entirely doubtful, none of which (fortunately) we have to reason on. **Therefore the true logic for this world is the calculus of Probabilities, which takes account of the magnitude of the probability which is, or ought to be, in a reasonable man's mind.**
> -James Clerk Maxwell (1850)

> "That's a bit of an anticlimax," Harry said. "You'd think there'd be some kind of more dramatic mental event associated with updating on an observation of infinitesimal probability -" Harry stopped himself. Mum, the witch, and even his Dad were giving him that look again. "I mean, with finding out that everything I believe is false."
> -Harry Potter-Evans-Verres, HPMOR.com, Chapter 2

Klassisessa päättelyssä aloitetaan oletuksista ja edetään niiden loogisiin seurauksiin. Pähkinänkuoressa:

- Kaikki ihmiset ovat kuolevaisia.
- Sokrates on ihminen.
- Täten Sokrates on kuolevainen.

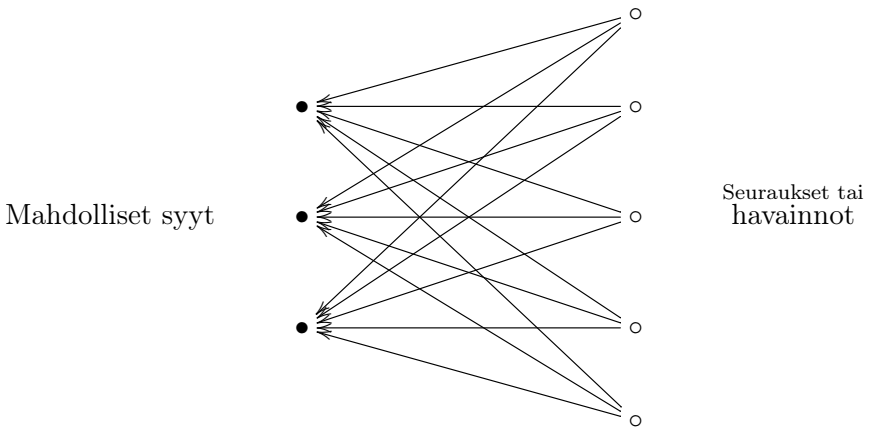
Kuvallisesti tilannetta voisi havainnoillistaa esimerkiksi seuraavasti.



Käytännön tilanteet ovat kuitenkin harvoin näin suoraviivaisia. Tyypillinen päättely voisi näyttää esimerkiksi seuraavalta:

- Maa on näköjään märkä.
- Päättelen että yöllä on satanut.
- (Vaikka tilanteen selittäisi myös yöllinen pikatulva tai naapuri joka on jostain syystä kastellut maata.)

Yleensä kuljemmekin siis päättelyissä ikäänkuin toiseen suuntaan. Teemme havaintoja, muodostamme näille mahdollisia selityksiä ja pidämme joitakin mahdollisista selityksiä toisia luultavampina. Kuvallisesti esitettynä tilanne näyttää seuraavalta.



Tilanne näyttää kuvan tasolla myös tieteellisen kokeen tekemiseltä; haluamme selvittää mikä useista mahdollisista hypoteeseistä on todennäköisin ja istuu parhaiten mittaamaamme dataan.

Mikä olisi sitten hyvä tapa verrata eri mahdollisuuksia? Coxin lauseen nojalla todennäköisyyslaskenta (eli Bayesiläinen päättely) antaa lähes[2] ainoan tavan suorittaa konsistenttia vertailua. Lähdetään seuraavaksi käymään läpi Coxin lauseen taustaa ja ideaa. Emme käy tässä läpi kaikkia yksityiskohtia, vaan lähestymistapamme on tässä vaiheessa luonteeltaan kuvailevampi. Uteliaan lukijan ohjaamme joko lukemaan asiasta lyhyesti lähteestä [Siv06, s.1–10] tai pitkän kaavan kautta lähteestä [Jay03].

Uskottavuusasteista

Tutkitaan tilannetta, jossa meillä on jokin systeemi jonka suhteen haluamme tehdä mahdollisimman hyviä ennustuksia ja päätelmiä. Tällainen systeemi voisi olla vaikka

- Aurinkokunnan fysikaalinen malli. (Haluamme saada mahdollisimman hyvän arvion Jupiterin massasta.)
- Kuurupiilo kaverin kanssa. (Haluamme löytää kaverimme mahdollismman nopeasti, eli haluamme tietää hänen luultavimmat piilopaikkansa.)
- Kryptattu viestikanava. (Haluamme tietää mahdollisimman paljon salakirjoitettujen viestien sisällöstä.)

Kaikissa systeemeissä meillä on aina myös jokin *konteksti*, jota merkitsemme tässä merkillä X. Konteksti kuvaa kaikkea (systeemin kannalta oleellista) taustatietoa mitä meillä on. Konteksti voi sisältää esimerkiksi

- Newtonin lait ja karkeita arvioita aurinkokunnan rakenteesta.
- Tietoa piilossa olevan henkilön koosta ja hänen piilopaikkapreferenssistään.
- Suomenkielisen tekstin kirjainfrekvenssejä ja salakirjoitusalgoritmin rakenteen.

Konteksti yleensä laajenee kun opimme lisää ympäristöstämme. Kontekstiin voi tulla esimerkiksi uusia mittauksia Jupiterin kiertoradasta, tieto siitä että kaverisi ei mennyt sisälle piiloon tai että salausalgoritmin käyttäjä on tehnyt protokollassaan virheen.

Tämän lisäksi meillä on erilaisia *mahdollisuuksia*, (tai tapahtumia), joita merkitään tässä A, B, C, jne. Mahdollisuuksia voisivat esimerkiksi olla

- Jupiterin massa on yli miljoona tonnia.
- Piilossa oleva kaveri on joko kiven takana tai puussa.

[2] Coxin antamat aksioomat ovat useimmat kohtuullisesti hyväksyttävän tuntuisia, mutta tähän vaikuttaa kreikkalaiseen filosofiaan perustuva maailmankuvamme.

- Selkokielisen viestin ensimmäinen kirjain on “k”.

Uskottavuusaste on mittari, jolla *vertaillaan* sitä miten luultavia eri tapahtumat ovat keskenään. Tapahtuman A uskottavuusastetta kontekstissa X merkitään $\pi(A|X)$. Uskottavuusasteelta oletetaan seuraavia ominaisuuksia.

(C1) Uskottavuusasteita voi vertailla keskenään, eli tapahtumaa A voidaan pitää tapahtumaa B luultavampana kontekstissa X. Merkitään tätä $\pi(A|X) \geq \pi(B|X)$. (Koko uskottavuusasteen idea on vertailla asioita.) Lisäksi oletetaan, että tämä vertailu on *transitiivista*, eli jos $\pi(A|X) \leq \pi(B|X)$ ja $\pi(B|X) \leq \pi(C|X)$, niin myös $\pi(A|X) \leq \pi(C|X)$.

Tämä vaatimus estää kehäpäätelmät. Lisäksi teemme hieman teknisen, mutta hemmetin luonnollisen, oletuksen siitä että erilaisia tapahtumia on enintään kontinuumin verran. (Eli niitä saa olla äärettömästi, mutta ei naurettavan äärettömästi. Suurempia äärettömyyksiä tulee yleensä vastaan vaan tarkoituksenhakuisilla matemaatikoilla.) Tästä seuraa[3], että *voimme olettaa uskottavuusasteiden olevan reaalilukuja.*

(C2) Uskottavuusasteet sopivat yhteen matemaattisen logiikan kanssa toteuttaen seuraavat ehdot. (Kritiikkimme matemaattiste logiikkaa kohtaan ei ollu sen virheellisyydessä vaan riittämättömyydessä. Hyvän päättelysysteemin tulisi silti toimia samalla tavalla niissä spesifeissä tilanteissa jossa deduktiivinen logiikka on käyttökelpoista.)

(a) Jos tapahtuman uskottavuusaste voidaan päätellä useammalla eri tavalla, niin näiden tapojen tulee tuottaa sama uskottavuusaste.

(b) Kaikki taustainformaatio otetaan aina huomioon.

(c) Loogisesti yhtäpitävillä väitteillä on sama uskottavuusaste.

(C3) Voimme päätellä tapahtuman “ei A” (merkitään $\overline{A}$) luultavuuden kontekstissa X tapahtuman A luultavuudesta kontekstissa X. Tarkemmin ilmaistuna, on olemassa funktio $G\colon \mathbb{R} \to \mathbb{R}$ siten että

$$\pi(\overline{A}|X) = G(\pi(A|X)). \tag{1.2.1}$$

(C4) On olemassa funktio $F\colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, joka ei ole vakio kummankaan parametrinsa suhteen, siten että kaikilla tapahtumilla A ja B pätee

$$\pi(A \wedge B|X) = F(\pi(A|X \wedge B), \pi(B|X)). \tag{1.2.2}$$

Tässä $\wedge$ tarkoittaa “ja”, ja $\pi(A|X \wedge B)$ kuvaa tilannetta jossa olemme lisänneet kontekstiimme havainnon B.

[3] Tämä nojaa siihen, että jokainen hyvinjärjestetty joukko jossa on enintään kontinuumin verran alkioita uppoaa isomorfisesti avaruuteen $(\mathbb{R}, \leq)$.

> Tämä oletus on mahdollisesti raskain hyväksyä, sillä se on varsin tekninen. Suosittelen miettimään sen sisältöä hieman hartaammin, ja pohtimaan että minkälainen päättely- taikka uskomusjärjestelmä ei toteuttaisi tätä ehtoa.

Coxin lause kertoo, että näillä oletuksilla voimme uudelleenparametrisoida uskottavuusasteemme noudattamaan Kolmogorovin aksioomia. (Eli todennäköisyyslaskennan perusaksioomia.) Emme käy läpi kaikkia yksityiskohtia, mutta kuvailemme todistuksen suuria linjoja. Noudatamme pitkälti Sivian ([Siv06]) esitystä.

Ensialkuun huomaamme, että käyttämällä ehtoa (1.2.2) tapahtuman $A \wedge B \wedge C$ uskottavuuden arviointiin saamme

$$\begin{aligned}\pi(A \wedge B \wedge C|X) &= F(\pi(A \wedge B|X), \pi(C|A \wedge B \wedge X)) \\ &= F(F(\pi(A|X), \pi(B|A \wedge X)), \pi(C|A \wedge B \wedge X))\end{aligned}$$

ja vastaavasti

$$\begin{aligned}\pi(A \wedge B \wedge C|X) &= F(\pi(A|X), \pi(B \wedge C|A \wedge X)) \\ &= F(\pi(A|X), F(\pi(C|A \wedge B \wedge X), \pi(B|A \wedge X))).\end{aligned}$$

Ehdon (C2) a)-kohdan nojalla nyt saadaan siis

$$F(F(x, y), z) = F(x, F(y, z)).$$

Tässä kohtaa suoritamme ison välivaiheiden ohittamisen ja toteamme, että koska F toteuttaa ylläolevan yhtälön kaikilla $x, y, z \in \mathbb{R}$, niin F on välttämättä muotoa

$$F(x, y) = f^{-1}(f(x) + f(b)),$$

missä $f\colon \mathbb{R} \to \mathbb{R}$ on kääntyvä kuvaus.[4]

Seuraavaksi määrittelemme $\phi = f \circ \pi$ ja huomaamme että koska f on kääntyvä kuvaus, niin ehdon (C3) nojalla niin on olemassa $g\colon \mathbb{R} \to \mathbb{R}$ siten, että

$$\phi(\overline{A}|X) = g(\phi(A|X)).$$

Lisäksi näemme, että nyt pätee

$$\phi(A \wedge B|X) = \phi(A|B \wedge X) + \phi(B|X).$$

Teemme toisen ison välivaiheiden ohittamisen, ja toteamme että rajoittumalla tiettyihin minimalistisiin tilanteisiin[5] voidaan funktiolle ϕ johtaa

[4] Tämä on kovasti epätriviaali huomio, kts. esim [Siv06, Appendix C].

[5] Päättelytekniikkamme tulee toimia kaikissa tilanteissa, joten sen pitää erityisesti toimia kyseisessä minimalistisessa tilanteessa, joka edelleen sisältyy lähes kaikkiin yleisimpiin systeemeihin. Joten koska funktion g pitää toteuttaa ominaisuus minimitilanteessa, pitää sen toteuttaa se myös sen sisältävissä rakenteissa. Kts. jälleen kerran [Siv06].

seuraava ominaisuus:

$$x + g(g(y) - x) = y + g(g(x) - y) \quad \text{kaikilla } x, y \in \mathbb{R}.$$

Tästä edelleen epätriviaalisti seuraa, että funktion g on oltava muotoa

$$g(z) = \gamma^{-1} \log(1 - e^{\gamma z}),$$

missä γ on jokin nollasta poikkeava vakio.

Nyt viimein voimme määritellä $\text{prob}(\cdot) = \exp(\gamma\phi(\cdot))$, ja huomaamme suoralla laskulla että

$$\begin{aligned}
\text{prob}(A|X) + \text{prob}(\overline{A}|X) &= \text{prob}(A|X) + \exp(\gamma\phi(\overline{A}|X)) \\
&= \text{prob}(A|X) + \exp(\gamma g(\phi(A|X))) \\
&= \text{prob}(A|X) + \exp(\gamma\gamma^{-1} \log(1 - e^{\gamma\phi(A|X)}) \\
&= \text{prob}(A|X) + 1 - e^{\gamma\phi(A|X)} \\
&= e^{\gamma\phi(A|X)} + 1 - e^{\gamma\phi(A|X)} \\
&= 1.
\end{aligned}$$

Vastaava suora lasku näyttää, että

$$\text{prob}(A \wedge B|X) = \text{prob}(A|X \wedge B)\,\text{prob}(B|X).$$

Täten alkuperäisestä uskottavuusasteesta $\pi(\cdot)$ uudelleenparametrisoitu versio

$$\text{prob}(\cdot) = \exp(\gamma\phi(\cdot)) = \exp(\gamma f(\pi(\cdot)))$$

toteuttaa Kolmogorovin aksioomat todennäköisyydelle.

Pidemmekin jatkossa annettuna, että todennäköisyyteen perustuva Bayesiläinen päättely on luonnollinen malli tehdä informatiivisia päätelmiä.

1.2.2 Päättelyä käytännön tilanteissa

Coxin lauseen nojalla Kolmogorovin aksioomat antavat hyvän, elleivät suorastana parhaan (ja ainoan) tavan arvioida tapahtumien uskottavuutta annetussa kontekstissa. Miltä tämä päättely sitten käytännössä näyttää? Kriittinen idea tiivistyy todennäköisyyslaskennasta tuttuun Bayesin kaavaan:

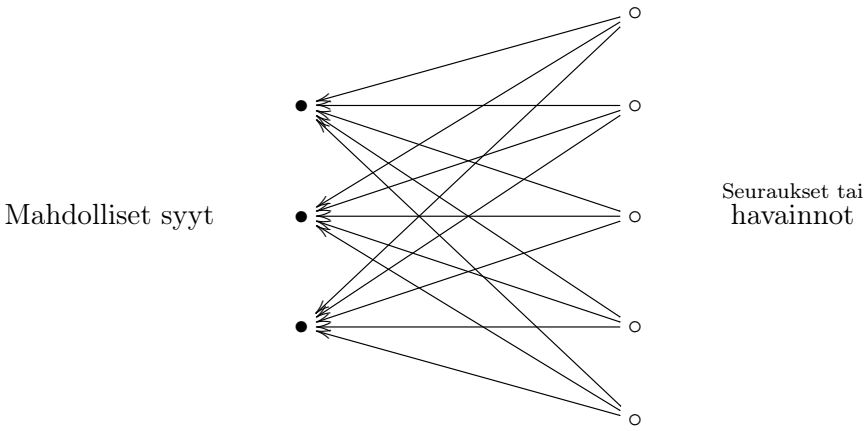
$$\text{prob}(X|Y, I) = \frac{\text{prob}(Y|X, I) \cdot \text{prob}(X|I)}{\text{prob}(Y|I)}$$

Tässä on siis kyseessä yhteisjakauma $(\Omega_1 \times \Omega_2, F_1 \times F_2, P)$ missä X ja Y ovat satunnaismuuttujia todennöisyysavaruuksissa (Ω_1, F_1, P_1) ja (Ω_2, F_2, P_2), missä P_1 ja P_2 ovat todennäköisyyden P marginaalitodennäköisyydet. Kaavoissa esiintyvä I kuvastaa meillä tilanteen kontekstia taikka tausta**I**nformaatiota.

Baysiläisen päättelyn taustalla on idea, että uskosi satunnaismuuttujan X laidasta muuttuu Bayesin kaavan kuvaamalla tavalla kun havainnoit/mittaat tapahtuman Y. Esimerkkinä voisi olla, että paras arvauksesi (jota kuvastaa satunnaismuuttuja X) kaverisi piilopaikasta muuttuu saatuasi tietoosi että hän ei ole piilossa sisällä (tapahtuma Y). Bayesin kaavan voi myös vihjailevasti kirjoittaa seuraavaan muotoon.

$$\text{prob}(\text{hypoteesi}|\text{data}, I) = \frac{\text{prob}(\text{data}|\text{hypoteesi}, I) \cdot \text{prob}(\text{hypoteesi}|I)}{\text{prob}(\text{data}|I)}$$

Bayesin kaavan voidaan siis tulkita kertovan myös siitä, miten uskottavuusasteesi mahdollisiin eri hypoteeseihin muuttuu saatuasi uutta (mittaus)dataa tilanteesta. Toistamme aikaisemman kuvan korostaaksemme, että tähän juuri pyrimme:



Todennäköisyyslaskenta on siis eri mahdollisuuksien uskottavuuksien vertailuun sopiva koneisto Coxin lauseen mukaan, ja Bayesin lause antaa konkreettisen työkalun tähän vertailuun.

Huomaa vielä, että mikäli vertailemme eri hypoteeseja toisiinsa, meitä kiinnostaa vain mille hypoteesille Bayesin kaava antaa maksimaalisen arvon, eikä nimittäjän termi vaikuta tähän. Sen voi siis (joissain tilanteissa) jättää huomiotta, jolloin voimme kirjoittaa

$$\text{prob}(\text{hypoteesi}|\text{data}, I) \propto \text{prob}(\text{data}|\text{hypoteesi}, I) \cdot \text{prob}(\text{hypoteesi}|I).$$

Klassinen Monty Hall -ongelma

Analysoidaan Bayesilaisella päättelyllä seuraavaa tilannetta, joka tunnetaan Monty Hall -ongelmana.

Esimerkki 6. Olet mukana kisailuohjelmassa jossa edessäsi on kolme suljettua ovea. Yhden takana on palkinto, kahden muun takana ei mitään. Saat valita vapaasti oven, mutta ennenkuin avaat sen, kisan juontaja avaa kahdesta muusta ovesta sellaisen, jonka takana ei ole palkintoa. (Hän tietää

palkinnon sijainnin.) Tämän jälkeen sinulle tarjotaan mahdollisuus vaihtaa ovea. Kannattaako sinun vaihtaa?

Kilpailussa alkuperäinen valinta tehdään ennen kuin saamme mitään informaatiota palkinnon sijainnista, joten voimme olettaa että valitsemme aluksi oven 1. (Muut tapaukset voi käsitellä erikseen, tai huomata että tässä vaiheessa voimme numeroida ovet päässämme uudestaan.)

Muodostetaan seruaavaksi tilannetta kuvaava todennäköisyysjakauma. Tilannetta varten muodostamme yhteisjakauman $(\mathcal{H} \times \mathcal{A}, \mathcal{P}(\mathcal{H}) \times \mathcal{P}(\mathcal{A}), P)$, missä $\mathcal{H} = (H_1, H_2, H_3)$ kuvaa palkinnon sijainnin vaihtoehtoja (H_i on alkeistapahtuama "palkinto on oven i takana") ja $\mathcal{A} = (A_1, A_2, A_3)$ kuvaa juontajan avaamaa ovea (A_i on alkeistapahtuma "juontaja avaa oven i"). Muodostetaan seuraavaksi todennöisyysfunktio P.

Emme tiedä aluksi mitään palkinnon sijainnista, joten meidän on luontevaa asettaa tapauksien H_i marginaalitodennäköisyydet samoiksi: $P(H_1) = P(H_2) = P(H_3) = \frac{1}{3}$. Koska tiedämme, että juontaja avaa aina oven jonka takana palkinto ei ole, saamme lisäksi seuraavat ehdolliset todennäköisyydet. (Huomaa, että juontaja ei ikinä avaa ovea 1, koska olemme sen alustavasti valinneet.)

$$P(A_1|H_1) = 0 \qquad P(A_2|H_1) = \tfrac{1}{2} \qquad P(A_3|H_1) = \tfrac{1}{2}$$

$$P(A_1|H_2) = 0 \qquad P(A_2|H_2) = 0 \qquad P(A_3|H_2) = 1$$

$$P(A_1|H_3) = 0 \qquad P(A_2|H_3) = 1 \qquad P(A_3|H_3) = 0.$$

Näiden tietojen avulla voimme laskea todennäköisyysfunktion P arvot kaavalla $P(A_i, H_j) = P(A_i|H_j)P(H_j)$. Saamme

$$P(A_i, H_j) = \begin{cases} \frac{1}{6}, & \text{kun } i \in \{2,3\}, j = 1 \\ \frac{1}{3}, & \text{kun } (i,j) \in \{(2,3),(3,2)\} \\ 0, & \text{muulloin.} \end{cases}$$

(Emme tosin tarvitse koko jakaumaa Bayesin kaavan ansiosta.)

Nyt voidaan laskea

$$P(H_j|A_2) = \frac{P(A_2|H_j)P(H_j)}{P(A_2)} = \frac{P(A_2|H_j)\frac{1}{3}}{\frac{1}{2}}$$
$$= \frac{2}{3}P(A_2|H_j) = \begin{cases} \frac{1}{3}, & \text{kun } i = 1 \\ 0, & \text{kun } i = 2 \\ \frac{2}{3}, & \text{kun } i = 3. \end{cases}$$

Vastaavalla laskulla saadaan

$$P(H_j|A_3) = \begin{cases} \frac{1}{3}, & \text{kun } i = 1 \\ \frac{2}{3}, & \text{kun } i = 2 \\ 0, & \text{kun } i = 3. \end{cases}$$

Nähdään siis, että kussakin tilanteessa toinen ovi kuin alkuperäinen valinta on todennäköisempi kuin alkuperäinen valinta. Vaihtaminen siis kannattaa!

Tämä tulos voi useasti vaikuttaa epäintuitiiviselta. Suosittelen jotain seuraavista:

- Pelaa pelia opiskelijatoverisi kanssa 20 kertaa kahdella eri strategialla ja kirjaa voittojen määrä. Välineeksi sopii esimerkiksi kolme pelikorttia.
- Koodaa itsellesi pieni ohjelma joka simuloi pelaamista miljoona kertaa ja kerää tulokset.
- Mieti tilannetta jossa on alunperin miljoona ovea. Valitset yhden, jonka jälkeen juontaja avaa 999 998 ovea. Kannattaako vaihtaa?

Varioitu Monty Hall -ongelma

Esimerkki 7. Kuten yllä, mutta tällä kertaa juontaja päättää kapinoida kisailuohjelman tuottajaa vastaan. (Hän on katkeroitunut koska hänen ideoitaan ohjelman kehittämisestä ei kuunneltu.) Valittuasi alkuperäisen oven hän kaivaa taskustaan kolmisivuisen nopan, heittää sitä ja valitsee nopan silmäluvun perusteella minkä oven hän aukaisee. Kyseisen oven takana ei ole palkintoa. Kannattaako sinun nyt muuttaa valintaasi?

Analysoidaan taas. Otetaan nyt uusi yhteisjakauma, jota merkitsemme (V, A^*), muotoa $(V \times A^*, \mathcal{P}(V) \times \mathcal{P}(A^*), P)$, missä

$$V = (V_1, V_2, V_3), \quad A^* = \left(A_1^+, A_1^-, A_2^+, A_2^-, A_3^+, A_3^-\right).$$

Lasketaan suoraan yhteisjakauman todennäköisyysfunktio. Arvot on kirjattu seuraavaan taulukkoon.

	A_1^+	A_1^-	A_2^+	A_2^-	A_3^+	A_3^-
V_1	0	$\frac{1}{9}$	$\frac{1}{9}$	0	$\frac{1}{9}$	0
V_2	$\frac{1}{9}$	0	0	$\frac{1}{9}$	$\frac{1}{9}$	0
V_3	$\frac{1}{9}$	0	$\frac{1}{9}$	0	0	$\frac{1}{9}$

Marginaalitodennäköisyyksiksi saadaan $P(\mathcal{V} = V_i) = \frac{1}{3}$, $i = 1, 2, 3$, ja marginaalijakauman $\mathcal{A}$ todennäköisyydet ovat

$$\left(\frac{2}{9}, \frac{1}{9}, \frac{2}{9}, \frac{1}{9}, \frac{2}{9}, \frac{1}{9}\right).$$

Nyt voidaan taas laskea

$$\begin{aligned} P(V_j|A_2^-) &= \frac{P(A_2^-|V_j)P(V_j)}{P(A_2^-)} = \frac{P(A_2^-|V_j)\frac{1}{3}}{\frac{1}{3}} \\ &= \frac{2}{3}P(A_2^-|V_j) = \begin{cases} \frac{1}{2}, & \text{kun } i = 1 \\ 0, & \text{kun } i = 2 \\ \frac{1}{2}, & \text{kun } i = 3. \end{cases} \end{aligned}$$

Vastaavalla laskulla saadaan

$$P(V_j|A_3^-) \begin{cases} \frac{1}{2}, & \text{kun } i = 1 \\ \frac{1}{2}, & \text{kun } i = 2 \\ 0, & \text{kun } i = 3. \end{cases}$$

Tässä tilanteessa vaihtamisella ei siis ole väliä! Tilanteista voi lukea lisää MacKayn kirjasta, [Mac03, s.57, Exercise 3.8 & 3.9]

Miten paljon informaatiota tilanteessa liikkuu?

Ensimmäisessä tilanteessa epävarmuuttamme palkinnon sijainnista kuvasi tasajakauma $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$, jonka entropia on $\log_2(3) \approx 1.58$. Tutkitaan tilannetta, jossa aluksi valitsimme oven yksi, ja juontaja avasi oven kaksi. Aiemmista laskuista näemme, että $P(\mathcal{A} = A_2) = \frac{1}{2}$, joten tapahtuman "ovi kaksi aukaistiin" informaatiosisältö on $h(\frac{1}{2}) = 1$. Tapahtuman jälkeen päädyimme siihen, että uskoamme palkinnon sijainnista kuvaa jakauma $(\frac{1}{3}, 0, \frac{2}{3})$. Tämän jälkimmäisen (posteriori) jakauman entropia on noin 0.92. Koska $1.58 - 1 = 0.58 < 0.92$, niin kaikki informaatio juontajan oven avauksesta ei siirtynytkään tiedoksi palkinnon sijainnista. Minne nuo 0.34 bittiä informaatiota päätyivät? Tulemme palaamaan aiheeseen myöhemmin, lyhyt vastaus on että "kadonnut" informaatio liittyy satunnaismuuttujien $\mathcal{V}$ ja $\mathcal{A}$ yhteisinformaatioon. Lisäksi huomaamme että yhteisjakauman entropialle voi laskea $H(\mathcal{V}, \mathcal{A}) \approx 1.92$. Erityisesti siis juontajan tarjoama yhden bitin informaatio vähensi epävarmuuttamme yhteisjakauman tilasta täsmälleen yhden bitin verran.

Toisesta esimerkistä voi halutessaan laskea vastaavia tunnuslukuja.

Katso kuitenkin näihin liittyen varoittava esimerkki 8

Luku 2

Entropia ja informaatiosisältö

Tässä osiossa syvennetään aiempia huomioita entropian sekä informaatiosisällön ominaisuuksista. Kertauksenomaisen tiivistyksen hengessä aiemmasta:

1. Käytämme todennäköisyysjakaumia useasti epävarmuuden kuvailuun emmekä niinkään satunnaisuuden kuvailuun.

2. Coxin lauseen nojalla tämä on todellä hyvä idea.

3. Todennäköisyysjakauma kuvaa epävarmuuttamme, ja todennäköisyysjakauman entropia epävarmuutemme määrää.

4. Tapahtuman $A = a_0$ informaatiosisältö $h(A) = -\log_2(P(A = a_0))$ kuvaa sitä, paljon tietoa tapahtuman havainnointi meille antaa. Entropia on informaatiosisällön odotusarvo.

5. Huomaa kolikon kaksi puolta. Muodostamme epävarmuuttamme kuvaamaan todennäköisyysjakauman jonka entropia, eli keskimääräinen informaatiosisältö, kuvaa epävarmuutemme määrää. Kun havainnoimme tapahtuman (tai tutkimme paljonko annettu koejärjestely, esimerkiksi punnitus, meille kertoo) laskemme tapahtuman informaatiosisällön (tai koejärjestelyn entropian) joka kertoo[1] meille paljonko epävarmuutemme vähenee (keskimäärin). Tilanne toivon mukaan selkeytyy ainakin hieman tämän luvun aikana.

Tässä kappaleeessa kaikki todennköisyysjakaumamme ovat muotoa $(\omega, \mathcal{P}(\Omega), P)$, missä Ω on äärellinen joukko. merkitsemmekin usein todennäköisyysjakaumaa $((a_1, a_2, \ldots, p_k), (p_1, p_2, \ldots, p_k))$, millä tarkoitamme siis että $P(a_j) = p_j$. (Muista, että äärellisessä todennäköisyysjakaumassa alkeistapauksien todennäköisyydet määräävät koko todennäköisyysfunktion.

[1] Ei aina suoraan, katso esimerkiksi viime luvun Monty Hall -ongelmien informaatiolaskut.

Monisteen merkintöihin on livahtanut epäkonsistenssia. Tapahtuman "A ja B" todennäköisyyttä merkitään ajoittain $P(A \wedge B)$, ajoittain $P(A, B)$. Tarkoittavat samaa asiaa. Pahoittelut.

Caveat Lector: Tutkittaessa äärellistä todennäköisyysjakaumaan jossa P on vakio, entropian laskeminen palautuu kysymykseen että "montako kertaa joukko pitää puolittaa jotta jäljelle jää vain yksi alkio". Toisin sanoen, "kuinka monta kyllä/ei-kysymystä pitää kysyä että yksittäinen alkio määrittyy". Monimutkaisemmassa tilanteessa, kuten aiemmassa punnitusongelmassa, voidaan tehdä monimutkaisempia mittauksia kuin kyllä/ei-kysymyksiä (esim. vasen/oikea/tasapaino), jolloin on luontevaa kysyä montako kertaa joukko pitää jakaa kolmeen ennen yksittäisen alkion eriytymistä. Tässäkin entropia tarjoaa ratkaisun laskea vaadittavien kysymyksien määrä. Meillä on kuitenkin useammin käsillä tilanne, missä epävarmuuttamme ei kuvaa tasajakauma. Tällöin heuristisessa mielessä kysymys "kuinka monta kyllä/ei-kysymystä pitää kysyä..." muuttuu muotoon "kuinka monta kyllä/ei-kysymystä pitää **keskimäärin** kysyä...".

Tarkemmin sanottuna, mikäli epävarmuuttasi kuvaa jokin TN-jakauma (Ω, F, P) ja havainnoit tapahtuman A jonka informaatiosisältö on $h(A)$, ei tämä aina tarkoita että päivitetyn jakaumasi entropiaa voi päätellä tai laskea alkuperäisen jakauman entropiasta sekä luvusta $h(A)$. Voit kuitenkin laskea, että mikäli aiot tehdä mittauksen X jonka mahdollisuudet ovat A ja $\overline{A}$, niin

$$H_\Omega = H(X) + P(X = A)H(\Omega|X = A) + P(X = \overline{A})H(\Omega|X = \overline{A}),$$

missä $H(\Omega|X = A)$ tarkoittaa että laskemme entropiaa uudelle todennäköisyyjakaumalle $(\Omega \cap A, F', P(\cdots|A))$. (Puhumme tästä nk. *ehdollisesta entropiasta* kohta lisää.)

Varoittava esimerkki:

Esimerkki 8. Kuvatkoon epävarmuuttamme jostakin tilanteesta jakauma, jossa on viisi alkeistapausta, $a_1, \ldots, a_5$, joilla on todennäköisyydet $p_1, \ldots, p_5$ siten, että $p_1 = \frac{1}{2}$ ja $p_2 = \ldots = p_5 = \frac{1}{8}$. Tämän jakauman entropiaksi saadaan

$$H_{\text{alku}} = \frac{1}{2}\log_2(2) + 4\frac{1}{8}\log_2(8) = \frac{1}{2} + 3\frac{1}{2} = 2.$$

Tapahtuman $\{a_2, \ldots, a_5\}$ todennäköisyys on $4 \cdot \frac{1}{8} = \frac{1}{2}$, eli kyseisen tapahtuman informaatiosisältö on $h(\frac{1}{2}) = 1$. Tällä tapahtumalla ehdollistettu uusi todennäköisyysjakauma on

$$((a_1, a_2, a_3, a_4, a_5), (0, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4})).$$

Tälle jakaumalle saadaan entropiaksi $4 \cdot \frac{1}{4} \cdot \log_2(4) = 2$.

Toisaalta tapahtuman $\{a_1\}$ todennäköisyys myöskin $\frac{1}{2}$, eli kyseisen tapahtuman informaatiosisältö on $h(\frac{1}{2}) = 1$. Tällä tapahtumalla ehdollistettu uusi todennäköisyysjakauma on

$$((a_1, a_2, a_3, a_4, a_5), (1, 0, 0, 0, 0)).$$

Tälle jakaumalle saadaan entropiaksi 0.

Huomaammekin, että informaation saaminen tilanteessä pienensi epävarmuuttamme, mutta epävarmuuden määrä voi yksittäisessä tapauksessa pienentyä enemmän tai vähemmän kuin havaitun tapahtuman bittimäärä.

Kuitenkin, huomaa että kahden mahdollisen tapahtuman todennäköisyyksillään painotettu keskiarvo on

$$\frac{1}{2} \cdot 2 + \frac{1}{2} \cdot 0 = 1,$$

eli keskimäärin saamme yhden bitin tietoa.

Eli siis: **yleisessä tilanteessa missä epävarmuutesi on aluksi H ja saat tilanteeseen liittyen k bittiä informaatiota, voit arvioita uuden epävarmuutesi määrää – muutta vain keskiarvoisesti.**

2.1 Perusosa

Käydään läpi lisää entropian perusominaisuuksia ja määritellään yhteisinformaatio.

2.1.1 Informaatiosisällön ja entropian karakterisoivia ominaisuuksia

Määrittelimme aikaisemmin tapahtuman A informaatiosisällön kaavalla

$$h(P(A)) = \log_2\left(\frac{1}{P(A)}\right).$$

Infomraatiosisällöllä on seuraavia hyödyllisiä ominaisuuksia.

Lemma 2.1.1. *Funktiolle $h\colon (0,1] \to \mathbb{R}$ pätee, että:*

1. *Se on jatkuva ja vähenevä funktio.*
2. *Kaikilla $x, y \in (0,1]$ pätee $h(xy) = h(x) + h(y)$.*

Todistus. Seuraa suoraan logaritmin perusominaisuuksista. □

Mahdollisesti kiinnostavampi huomio on seuraava.

Lause 2.1.2. *Olkoon $f\colon (0,1] \to \mathbb{R}$ funktio s.e.*

1. *Se on jatkuva ja vähenevä funktio.*
2. *Kaikilla $x, y \in (0,1]$ pätee $f(xy) = f(x) + f(y)$.*

Tällöin funktio f on muotoa $f(x) = \log_r(1/x)$ jollakin $r > 0$.

Todistus. Demotehtävä. □

Vastaavasti huomataan, että entropialle pätevät seuraavat ominaisuudet.

Lemma 2.1.3. *Todennäköisyysjakaumassa $((a_1, \ldots, a_k), (p_1, \ldots, p_k))$ Shannonin enropiakuvaukselle*

$$H(p_1, \ldots, p_n) = -\sum_{j=1}^{n} p_j \log_2(p_j)$$

on voimassa seuraavat ominaisuudet.

1. *Kuvaus H on jatkuva kunkin parametrinsa suhteen.*
2. *Jos merkitään*

$$A(k) = H(\underbrace{\frac{1}{k}, \ldots, \frac{1}{k}}_{k \text{ kpl}})$$

niin $A(k) \geq A(m)$ aina kun $m \geq k$.

3. *Jos jakauman alkeistapaukset jaetaan erillisiksi joukoiksi* $A_1, \ldots, A_j$ *ja muodostetaan uudet jakaumat* $X := ((A_1, \ldots, A_j), (P(A_1), \ldots, P(A_j))$ *sekä* $Y_j := (A_j, P(\cdot|A_j))$*, niin entropialle pätee*

$$H(p_1, \ldots, p_n) = H(X) + \sum_{i=1}^{j} P(A_j)H(Y_j).$$

Todistus. Kohdat 1 ja 2 ovat suoraviivaisia. Kohdassa kolme todistettiin luennolla tilanne $\Omega = \{a_1, \ldots, a_5\}$, $A_1 = \{a_1, a_2\}$, $A_2 = \{a_3, a_4, a_5\}$, joka on suora lasku ja demonstroi hyvin yleisen tilanteen. Todistus lisätään tänne mahdollisesti myöhemmin. □

Myös entropialle pätee että nämä ominaisuudet karakterisoivat sen.

Lause 2.1.4 (Shannon 1946)**.** *Funktio joka toteuttaa ylläolevat ehdot on välttämättä muotoa*

$$-\sum_{j=1}^{n} p_j \log_r(p_j)$$

jollakin $r > 0$*.*

Todistus. Torstain luennolla. □

2.1.2 Yhteisinformaatio

Seuraavaksi perehdytään yhteisinformaation käsitteeseen. Aloitetaan muutamalla havainnoillistavallae esimerkillä.

Muutama esimerkki

`Kuurupiiloesimerkit ovat huollossa.`

Toisiinsa liityvät satunnaismuuttujat

Olkoon (X, Y) yhteisjakauma. Todennköisyyslaskennassa määritellään, että muuttujat ovat riippumattomat mikäli

$$P(A \wedge B) = P(A)P(B).$$

Tällaisessa tilanteessa huomaamme, että havainnon B tekeminen jälkimmäisestä satunnaismuuttujasta ei vaikuta ensimmäiseen satunnaismuuttujaan. Jälkimmäinen muuttuja ei siis välitä informaatiota ensimmäisen satunnaismuuttujan tilasta.

> "Information theory 101,"the boy said in a lecturing tone. "Observing variable X conveys information about variable Y, if and only if the possible values of X have different probabilities given different states of Y."
>
> Harry Potter-Evans-Verres, HPMOR.com

Klassinen tapa mitata satunnaismuuttujien yhteyttä on korrelaatiokerroin. Korrelaatiokerroin kuitenkin mittaa lähinnä lineaarista yhteyttä kahden satunnaismuuttujan välillä. Tarkempi mittari on *yhteisinformaatio*.

Huomataan alkuun että yhteisjakaumassa, jossa muuttujan ovat riippumattomia, pätee

$$\begin{aligned}
H(X,Y) &= \sum_{a_i,b_j} P(a_i,b_j)\log_2\left(\frac{1}{P(a_i,b_j)}\right) \\
&= -\sum_{a_i,b_j} P(a_i)P(b_j)\log_2\left(P(a_i)P(b_j)\right) \\
&= -\sum_{a_i,b_j} P(a_i)P(b_j)(\log_2(P(a_i)) + \log_2(P(b_j))) \\
&= -\left(\sum_{a_i} P(a_i)(\log_2(P(a_i))\sum_{b_j} P(b_j)\right) - \left(\sum_{b_i} P(b_i)(\log_2(P(b_i))\sum_{a_j} P(a_j)\right) \\
&= H(X) + H(Y).
\end{aligned}$$

Riippumattomilla satunnaismuuttujilla yhteisjakauman enropia on siis marginaalijakaumien entropioiden summa. Tästä saadaan tapa mitata jakaumien riippuvuutta tutkimalla paljonko nämä entropiat poikkeavat toisistaan.

Määritelmä 2.1.5. Yhteisjakaumassa (X,Y) kahden satunnaismuuttujan *yhteisinformaatio* on

$$I(X;Y) = H(X) + H(Y) - H(X,Y).$$

Yhteisinformaation voi määritellä myös seuraavalla lähestymistavalla.

Määritelmä 2.1.6. Olkoon (X,Y) yhteisjakauma. Määritellään

$$H(X|Y=b_j) = -\sum_{a_i} P(X=a_i|Y=b_j)\log_2(P(X=a_i|Y=b_j)).$$

Huomaa että tässä ehdollistetaan tapahtumalla.

Tulkitaan nyt $b_j \mapsto H(X|Y = b_j)$ satunnaismuuttujana ja määritellään sen odotusarvona

$$\begin{aligned} H(X|Y) &= E[H(X|Y = b_j)] \\ &= \sum_{b_j} P(Y = b_j) H(X|Y = b_j) \\ &= \dots \\ &= \sum_{a_i, b_j} P(X = a_i, Y = b_j) \log_2 \left(\frac{1}{P(X = a_i | Y = b_j)} \right). \end{aligned}$$

Tätä kutsutaan satunnaismuuttujan X ehdolliseksi entropiaksi satunnaismuuttujan Y suhteen.

Lemma 2.1.7. *Nyt pätee*

$$H(X, Y) = H(X) + H(Y|X) = H(Y) + H(X|Y),$$

josta saadaan

$$I(X; Y) = H(X) - H(X|Y) = H(Y) - H(Y|X).$$

Todistus. Sivuutetaan toistaiseksi. □

Huomautamme, yhteisinformaation voi laskea myös

$$I(X; Y) = D_{KL}(P(x, y) || P(x)P(y)),$$

missä $D_{KL}(\cdot||\cdot)$ on torstaina käsiteltävä Kullback-Leibler -divergenssi.

Esimerkki 9.

1. Aikaisemmassa esimerkissä golfpallosta ja jalkapallosta laskimme yhteisjakauman entropiaksi 2.8. Toisaalta pallon valinnan marginaalijakauma oli $(0.5, 0.5)$, eli sen enropia on 1 bitti, ja rasian valinnan marginaalijakauma oli $(1/3, 1/3, 1/6, 1/6, 1/6, 1/6)$, jonka enropia on 2.25. Täten yheisinformaation määrä on $2.8 + 1 - 2.25 = 1.55$ bittiä.

2. Klassisessa Monty Hall-ongelmassa laskimme, että yhteisjakauman $(\mathcal{V}, \mathcal{A})$ entropia oli 1.92. Toisaalta jakauman $\mathcal{V}$ (missä palkinto on) entropia oli 1.58 ja jakauman $\mathcal{A}$ (minkä oven juontaja avaa) entropia oli 1. Tästä saadaan yhteisinformaatioksi $1.58 + 1 - 1.92 = 0.66$ bittiä.

3. Varioidussa Monty Hall-ongelmassa yhteisjakauman enropia on 3.17, kun taas marginaalijakaumien entropiat ovat 1.58 sekä 2.50. Nyt yhteisinformaatioksi saadaan $1.58 + 2.50 - 3.17 = 0.91$.

2.2 Syventävä laajennos

Tässä osiossa todistetaan, että aiemmin mainitut entropian ominaisuudet itse asiassa karakterisoivat entropiafunktion. Tämän jälkeen tarkastellaan hieman maksimientropian periaatetta TN-jakaumia valittaessa, ja törmätään jälleen kerran Kullback-Leibler -divergenssiin.

2.2.1 Entropian karakterisaatio

Entropiafunktioon H on tungettu tähän mennessä vaihteleva määrä parametreja. Jos kyseisen funktion formalismi ahdistaa, niin voi joko ajatella että meillä on itseasiassa kokoelma kuvauksia $H^j \colon \mathbb{R}^j \to \mathbb{R}$ joita kaikkia merkitsemme H, tai että määritellään

$$\mathcal{R} := \bigcup_{n\geq 2} \left\{ (x_1, \ldots, x_n) \in \mathbb{R}^n \mid \sum_{j=1}^{n} x_j = 1, x_j \geq 0 \right\}.$$

Tyyli vapaa.

Lause 2.2.1. *Olkoon $H \colon \mathcal{R} \to [0, \infty)$ funktio, jolle pätee seuraavat ominaisuudet.*

1. *Kaikilla k, kuvaus $p_1, \ldots, p_k \mapsto H(p_1, \ldots, p_k)$ on jatkuva kaikkien parametriensa suhteen.*

2. *Kaikilla k, kuvaus $A \colon \mathbb{N} \to \mathbb{R}$, missä*

$$A(k) := H(\underbrace{\frac{1}{k}, \ldots, \frac{1}{k}}_{k \text{ kpl}})$$

   on kasvava.

3. *Dekompositio-ominaisuus, eli jos $X := ((a_1, \ldots, a_n), (p_1, \ldots, p_n))$ on TN-jakauma, $\{A_j\}$ on jakauman X alkeistapauksien ositus erillisiin osajoukkoihin, ja $\mathcal{A}_j$ todennäköisyysjakauma muotoa $(A_j, P(\cdot|A_j))$ (eli jos $A_j = \{a_{k_1}, a_{k_2}, \ldots a_{k_j}\}$, niin*

$$P(a_{k_i}|A_j) = p_{k_i}/P(A_j) = \frac{p_{k_i}}{\sum_{j=k_i,\ldots,k_j} p_j})$$

   niin tällöin pätee että

$$H(X) = H(P(A_1), \ldots, P(A_k)) + \sum_j H(\mathcal{A}_j).$$

Todistus. Seuraavassa $s, t, m, n \geq 2$ ovat kaikki kokonaislukuja.

Huomataan ensin, että jos meillä on s^m alkeistapausta ja tasajakauma, niin kohdan 3 nojalla

$$\begin{aligned} A(s^m) &= A(s) + \sum_{j=1}^{s} \frac{1}{s} A(s^{m-1}) = A(s) + s\frac{1}{s}A(s^{m-1}) \\ &= A(s) + A(s^{m-1}) = A(s) + A(s) + A(s^{m-2}) \\ &= \ldots = \underbrace{A(s) + \ldots + A(s)}_{m \text{ kpl}} = mA(s). \end{aligned}$$

Kiinnitetään s, t sekä n ja valitaan m niin isoksi, että

$$s^m \leq t^n < s^{m+1}.$$

Koska A on oletuksen 2 nojalla aidosti kasvava, saadaan

$$\begin{aligned} & A(s^m) \leq A(t^n) < A(s^{m+1}) \\ \Leftrightarrow\ & mA(s) \leq nA(t) < (m+1)A(s). \end{aligned}$$

Jakamalla koko roska luvulla $nA(s)$ saadaan

$$\begin{aligned} & \frac{mA(s)}{nA(s)} \leq \frac{nA(t)}{nA(s)} < \frac{(m+1)A(s)}{nA(s)} \\ \Leftrightarrow\ & \frac{m}{n} \leq \frac{A(t)}{A(s)} < \frac{(m+1)}{n} = \frac{m}{n} + \frac{1}{n}. \end{aligned}$$

Toistamalla ylläoleva lasku, mutta kirjoittamalla $A(s)$ sijasta $\log(s)$, saadaan

$$\frac{m}{n} \leq \frac{\log(t)}{\log(s)} < \frac{(m+1)}{n} = \frac{m}{n} + \frac{1}{n}.$$

Erityisesti siis kolmioepäyhtälön nojalla

$$\left| \frac{A(t)}{A(s)} - \frac{\log(t)}{\log(s)} \right| = \left| \frac{A(t)}{A(s)} - \frac{m}{n} + \frac{m}{n} - \frac{\log(t)}{\log(s)} \right| \leq \frac{2}{n}.$$

Koska väite pätee mielivaltaisen suurilla n, saadaan siis

$$\begin{aligned} & \frac{A(t)}{A(s)} = \frac{\log(t)}{\log(s)} \\ \Leftrightarrow\ & A(t) = \frac{A(s)}{\log(s)} \log(t). \end{aligned}$$

Kiinnittämällä luvun s huomaamme siis, että kuvaus $t \mapsto A(t)$ on välttämättä muotoa $A(t) = K \log(t)$ jollakin vakiolla K.

Seuraavaksi, oletetaan että meillä on todennäköisyysjakauma jonka todennäköisyydet ovat muotoa $p_j = n_j/N$, missä $n_j, N \in \mathbb{N}$, $j = 1, \ldots, n$. Otetaan alkeistapauksien joukkoon $(a_1, \ldots, a_n)$ tasajakauma, ja jaetaan joukko osiin A_j siten että joukossa A_j on n_j alkiota. Nyt dekompositioehdon 3 nojalla

$$A(N) = H(p_1, \ldots, p_n) + \sum_j \frac{a_j}{N} A(a_j),$$

josta suoraviivainen lasku antaa, että

$$H(p_1, \ldots, p_n) + \sum_j \frac{a_j}{N} A(a_j) = -K \sum_j p_j \log(p_j)$$

jollakin vakiolla K.

Yleinen tilanne, jossa todennäköisyysjakauman ei koostu rationaaliluvuista seuraa ehdosta 1 ja siitä, että jatkuvan funktion arvot tiheässä joukossa määräävät funktion kaikki arvot. □

2.2.2 Maksimientropian periaate

Tilanteessa, jossa sinun pitää kuvata täyttä epävarmuuttasi TN-jakaumalla (eli esimerkiksi pitää valita yksi kolmesta ovesta ilman mitään tietoa minkä oven takaa löytyy palkinto) on luontevaa valita tasajakauma (eli ovien tilanteessa $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$). Tämä siistä syystä, että mikä tahansa muu jakauma 'väittää' että meillä on jotain tietoa palkinnon sijainnista.

Yleisemmin, todennäköisyysjakauman valinta on eräänlainen käänteinen ongelma; kun näemmä jakauman, voimme siitä kuvailla millaista epävarmuutta se kuvaa (esimerkiksi jakauma $(\frac{7}{9}, \frac{1}{9}, \frac{1}{9})$ kuvaisi, että olemma hyvin varmoja että palkinto on oven yksi takana, mutta jos se ei ole niin meillä ei ole tietoa onko ovi 2 vai 3). Nyt kuitenkin tilanteessa jossa meillä on joku kuvailu epävarmuudestamme, ja haluamme löytää TN-jakauman joka tilannetta parhaiten kuvaa.

Huomaa, että tilanne on hieman samankaltainen kuin Bayesin kaavaa käyttäessä, jolloin haluamme päivittää tuntemaamme jakaumaa uudella datalla. Tämän kappaleen aihe liittyy kuitenkin enemmän tilanteeseen, jossa emme tunne jakaumaa ollenkaan, ainoastaan joitain sen rakennetta rajoittavia tekijöitä.

Täyden epävarmuuden tilanteessa tasajakauma on luonnollinen valinta. Mutta usein tilanne on monimutkaisempi:

Esimerkki 10. Olkoon (X, Y) yhteisjakauma, josta tunnetaan ainoastaan marginaalijakaumat (eli tiedämme kahdesta tapahtumasta erikseen miten varmoja ajattelemme niiden olevan, mutta emme mitään niiden yhteysestä). Mikä jakauma meidän kannattaa valita yhteismuuttujalle?

Jos valitsemme yhteisjakaumalle todennäköisyydet jotka ovat muotoa $P(x, y) = P(x)P(y)$, eli oletamme jakaumat riippumattomiksi, niin emme väitä tietävämme mitään muuttujien X ja Y yhteydestä. Tämä on varsin luonnollinen valinta, sillä minkä tahansa muun jakauman valinta välittää tietoa näiden muuttujien korrelaatiosta.

Esimerkki 11. Olkoon (X, Y) yhteisjakauma, josta tunnetaan ainoastaan toinen marginaalijakauma, ja toisesta jakaumasta ainoastaan sen odotusarvo. (Eli tiedämme kahdesta tapahtumasta toisesta erikseen miten varmoja ajattelemme tapahtumien olevan, mutta toisesta jakaumasta vain keskimääräisen tilanteen.) Mikä jakauma kannattaa valita yhteismuuttujalle?

Tähän kysymykseen on hankala enää vastata ad hoc -ideoinnilla. Siirrytään siis apina-argumenttiin.

Oletetaan, että meilä on jokin tilanne, johon liittyvää epävarmuuttamme haluamme mallintaa TN-jakaumalla. Meillä on jonkin verran *laskettavissa olevia rajoitteita* jakaumalle, (kuten esimerkiksi odotusarvo tai ehto $p_1+p_4 \leq \sqrt{p_2}$) mutta ei niin paljoa dataa että osaisimme kiinnittää jakauman suoraan. Miten valita jakauma joka parhaiten kuvaa epävarmuuttamme?

Olkoon nyt $\Omega = \{x_1, \ldots, x_n\}$ aiottu alkeistapauksien joukko, johon haluamme löytää rajoitteisiimme sopivan jakauman. Oletetaan, että meillä on valtava määrä apinoita jotka tykkäävät heitellä kolikoita laatikoihin. Laitetaan apinoiden eteen laatikot $X_1, \ldots, X_n$ ja annetaan apinoille $N >> n$ kolikkoa joita saavat nakella laatikoihin. Kun kolikot on nakeltu ja kussakin laatikossa X_j on n_j kolikkoa, katsomme toteuttaako syntynyt jakauma $(a_1/N, a_2/N, \ldots, a_n/N)$ vaaditut laskettavissa olemme rajoitteet. Jos toteuttaa, vedämme viivan seinään ja kirjaamme jakauman ylös ja jos ei, niin emme. Tämän jälkeen tyhjennämme laatikot kolikoista ja pistämme apinat heittelemään kolikoita uudestaan. Hyvin monen toistokerran jälkeen katsomme mikä jakauma esiintyy kaikista useimmiten ja otamme sen kuvaamaan epävarmuuttamme. Koska apinoilla ei ole taustavaikutteita suosia joitain jakaumia toisia enemmän, eivät he yritä tuoda jakaumaan ominaisuuksia jotka edustavat tietoa mitä meillä ei tilanteesta ole. Tutkitaan seuraavaksi miten homma tehdään kynällä ja paperilla ilman apinoita.

Merkitään laatikoihin päätyneiden kolikoiden määrää $(a_1, \ldots, a_n)$ ja katsotaan millaiset jakaumat maksimoivat niiden frekvenssin:

$$\begin{aligned} F((a_j)) &= \frac{\text{miten monella tapaa saadaan } (a_j)}{\text{miten monta eri jonoa on}} \\ &= \frac{\binom{N}{a_1}\binom{N}{a_2}\cdots\binom{N}{a_n}}{n^N} = \frac{\frac{N!}{a_1!\cdot a_2!\cdot\ldots\cdot a_n!}}{n^N}. \end{aligned}$$

Meitä ei itseasiassa kiinnosta mikä maksimifrekvenssi on, vaan että millainen jakauma sen tuottaa. Koska logaritmi on kasvava funktio, niin $F((a_j))$ saavuttaa maksiminsa sillä jakaumalla jolla $\log(F((a_j)))$ saavuttaa maksiminsa.

Nyt päteekin

$$\begin{aligned}\log(F((a_j))) &= \log\left(\frac{\frac{N!}{a_1!\cdot a_2!\cdot\ldots\cdot a_n!}}{n^N}\right)\\ &= \log(N!) - \log(n^N) - \sum_{j=1}^{n}\log(a_n!).\end{aligned}$$

Stirlingin kaavan nojalla $\log(n!) \approx n\log(n) - n$, ja tämä approksimaatio on sitä parempi mitä suurempi n on kyseesä. Olettamalla, että kolikoita on aivan hillitön määrä, voidaan tätä approksimaatiota käyttää maksimoivan jonon etsimiseen:

$$\begin{aligned}\log(N!) - \log(n^N) - \sum_{j=1}^{n}\log(a_n!)&\\ \approx N\log(N) - N - \log(n^N) - \sum_j a_n\log(a_n) - \sum_j a_n&\\ = N\log(N) - \log(n^N) - \sum_j a_n\log(a_n).&\end{aligned}$$

Nyt edelleen

$$\begin{aligned}N\log(N) - \log(n^N) - \sum_j a_n\log(a_n)&\\ = -N\log(n) - \sum_j a_n\log(a_n) + \left(\sum_j a_n\right)\log(N)&\\ = -N\log(n) - \sum_j a_n\log(a_n) + \left(\sum_j a_n\right)\log(N)&\\ = -N\log(n) - \sum_j a_n\log(a_n/N)&\\ = -N\log(n) - N\sum_j \frac{a_n}{N}\log(a_n/N).&\end{aligned}$$

Huomataan että sekä n että N ovat vakioita, joten frekvenssin $F((a_j))$ maksimoi jakauma joka maksimoi termin

$$-\sum_j \frac{a_n}{N}\log(a_n/N) = H(\frac{a_1}{N},\ldots,\frac{a_n}{N}) = H(p_1,\ldots,p_n).$$

Täten siis annettujen reunaehtojen vallitessa, epävarmuuttamme parhaiten kuvaa jakauma joka maksimoi entropian.

Tilanteessa, jossa meillä on jotain etukäteistietoa jakaumasta, eli TN-jakauma $(q_1, \ldots, q_n)$, mutta olemme saaneet siihen uusia rajoitteita niin äskeinen argumentti korvataan tilanteella, jossa apinoille annetut laatikot ovat eri kokoisia jolloin niihin päätyy kolikkoja eri todennäköisyyksillä. Tällöin frekvenssi saa muodon

$$\begin{aligned} F((a_j)) &= \frac{\text{miten monella tapaa saadaan } (a_j)}{\text{miten monta eri jonoa on}} \\ &= \frac{\binom{N}{a_1}\binom{N}{a_2}\cdots\binom{N}{a_n}}{n^N} = \frac{N!}{a_1! \cdot a_2! \cdot \ldots \cdot a_n!} q_1 \cdot q_2 \ldots q_n. \end{aligned}$$

Ylläoleva lasku tuottaa silloin muuten saman tuloksen, paitsi että termi n^N korvataan tulolla $(q_1 \cdot q_2 \ldots q_n)^{-1}$, jolloin

$$\begin{aligned} \log(F((a_j))) &= -\sum_j \log(q_j) - \sum_j a_n \log(a_n/N) \\ &= -\sum_j a_n \log(\frac{a_n/N}{q_j}) \\ &= N^{-1} D_{KL}(P||Q). \end{aligned}$$

Eli tilanteessa jossa meillä on tietoa alkuperäisestä jakaumasta, apinat maksimoivat Kullback-Leiber divergenssin alkuperäisen jakauman suhteen.

Luku 3

Kolmogorov-kompleksisuus

Lähde: `https://xkcd.com/1155/`.

Tässä osiossa perehdytään informaatioon Kolmogorov-kompleksisuuden kautta. Kompleksisuuden formaali käsittely ei valitettavasti tämän kurssin esitiedoilla onnistu; meillä ei ole aikaa määritellä Turing-täydellisyyttä taikka käyttää paljoa aikaa automaattien teoriaan. Tämän osion tekstiin kannattaa suhtautua enemmän kuvailuna kuin matemaattisena teoriana. Suosittelemme voimakkaasti kurssin *TIEA241 – Automaatit ja Kieliopit* käymistä ja/tai Sipserin kirjan *Introduction to the theory of computation* lukemista.

3.1 Kolmogorov-kompleksisuus sekä Mystisk Box

Lähdetään hahmottamaan tilannetta formalisoimalla mitä tarkotiamme merkkijonoilla.

Määritelmä 3.1.1. *Aakkosto* on mikä tahansa epätyhjä äärellinen kokoelma alkioita, joita kutsumme symboleiksi;

$$\mathcal{A} := \{x_1, \ldots, x_k\}.$$

Aakkoston kokoa, eli alkioiden lukumäärää merkitään $|\mathcal{A}|$.

Esimerkki 12. Esimerkkejä aakkostoista.

1. Binääriaakkoset, $\mathcal{A}_b = \{0, 1\}$.
2. Numerot, $\mathcal{A} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$.
3. Pienet suomen kielen aakkoset:

$$\mathcal{A} = \{\text{a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z,å,ä,ö}\}$$

4. Suomen kielen aakkoset ja välimerkit

$$\mathcal{A} = \{\text{a}, \ldots, \text{ö}, \text{A}, \ldots, \text{Ö}, ., , , ”, !, ?, ; , :, −, –\}$$

5. Kaikki ASCII-merkit.

Määritelmä 3.1.2. Aakkoston $\mathcal{A}$ *merkkijonojen* (tai *sanojen*) joukko on

$$\mathcal{M}_{\mathcal{A}} = \{a_1 a_2 \cdots a_k | a_j \in \mathcal{A}, k \geq 0\}.$$

Merkitsemme *tyhjää sanaa* merkillä ϵ.

Täsmälleen pituutta N olevien merkkijonojen kokoelmaa merkitsemme $\mathcal{M}_{\mathcal{A}}^N$.

Esimerkki 13. Esimerkkejä merkkijonoista.

1. Binääriaakkostossa sanoja ovat kaikki binäärilukujonot, kuten 001, 110110011, tai 111111111111111.
2. Numeroiden aakkostossa $\mathcal{A} = \{0, \ldots, 9\}$ sanoja ovat esimerkiksi 103, 13 ja 000000000000000400000000000.
3. ASCII-aakkoston merkkijonoja ovat esimerkiksi “Tämä ei ole piippu.”, “Ota annetun luvun neliöjuuri.”, “Jav89jnöKJV!!!”, “En ole pelkkä merkkijono! Olen itsetietoinen matemaattinen objekti enkä halua joutua luentomonisteeseen!”, ja “hevonen”.

Kolmogorov-kompleksisuuden ideana on tutkia, mikä on lyhin mahdollinen kuvailu annetulle merkkijonolle. Kuvailun tyyppi pitää rajata, ja tyypillinen hyvä ratkaisu on rajata sallitut kuvailut nk. (Turing täydellisiin) *Turingin koneisiin.* Tällä kurssilla meillä ei ole mahdollisuuksia mennä Turingin koneen formaaliin määritelmään, mutta esimerkkejä Turing-täydellisistä Turingin koneista ovat mm. kaikki ohjelmointikielet (kun oletetaan että niillä on teoriassa ääretön muistikapasiteetti.)

Tilanteessa voi siis ajatella, että meillä on jokin ohjelmointikieli, esim Python3, ja kysymyksenä on että mikä on lyhin mahdollinen ohjelma joka tulostaa annetun merkkijonon. Mikäli ohjelmointi ei ole tuttua, niin voi myös ajatella että meillä on jonkinnäköinen algoritmi, resepti tai funktio joka muuttaa annettuja merkkijonoja toisiksi merkkijonoiksi. Funktioajattelussa kannattaa kuitenkin olla hieman tarkkana sillä;

- Ei ole aina ihan selvää mikä on oikea "määrittelyjoukko" (eli mitkä merkkijonot ovat "sallittuja ohjelmia").
- Ei ole aina ihan selvää minkä arvon funktio antaa syötteellä (emme tiedä kauanko algoritmi pyörii ennenkö se antaa vastauksen, tai että antaako se sitä ylipäätänsä.) Esimerkiksi algoritmista "(1) Aseta $n = 3$. (2) Katso onko n alkutekijöidensä summa; jos on niin kirjoita ylös luku n ja lopeta, muuten jatka. (3) Kasvata lukua n kahdella ja palaa kohtaan (2)." ei tiedetä pysähtyykö se koskaan. (Ei tiedetä onko olemassa parittomia täydellisiä lukuja.)

Mystisk Boxia U voi siis halutessaan ajatella funktiona $U\colon \mathcal{M}_{\mathcal{A}} \to (\mathcal{M}_{\mathcal{A}} \cup F)$, missä $F = \{\text{HYLÄTTY}, \text{EI PYSÄHDY}\}$. mutta tässä osiossa puhumme tukevan heuristisesti käsitteestä "Mystisk Box" joka ottaa sisäänsä syötteitä ja jollakin matemaattisen formaalilla tavalla tuottaa syötteistä ulos deterministisiä merkkijonoja.

Esimerkkejä käyttämästämme Mystisk Box -ideasta:

1. Sinä olet Mystisk Box kun luet annettuja ohjeita "Jaa annettua lukua toistuvasti kakkosella kunnes jakojäännökseksi tulee 1, ja kirjoita jäljellejäänyt luku paperille."
2. Python3 on Mystisk Box kun sille antaa komentoja kuten "print(13*'Abc')".

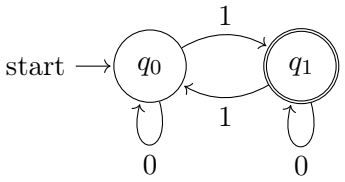
Seuraava *deterministinen äärellinen automaatti* ei ole Turing-täydellinen, mutta se on yksinkertainen Turingin kone joka kannattaa pitää mielessä kun pohtii miten mahdollinen Mystisk Box voisi toimia.

Määritelmä 3.1.3. *(Deterministinen) Äärellinen Automaatti* on viisikko $(Q, \Sigma, \delta, q_0, F)$, missä

1. $Q = \{q_1, \ldots, q_n\}$ on äärellinen joukko *tiloja*.
2. Σ on aakkosto.
3. $\delta\colon Q \times \Sigma \to Q \times (\Sigma \cup \{\epsilon\})$ on *siirtymäfunktio*.
4. $q_0 \in Q$ on alkutila.
5. F on *sallittujen lopputilojen* kokoelma.

Äärelliselle automaatille annetaan syöte joukosta $\mathcal{M}_\Sigma$, ja se lukee syötettään yksi symboli kerrallaan. Äärellinen automaatti aloittaa annetusta alkutilasta, ja siirtyy sitten saamansa syötteen perusteella askel kerrallaan uuteen tilaan siirtymäfunktion osoittamalla tavalla. Mikäli siirtymäfunktion arvo annetussa tilassa annetulla syötteellä on muotoa (q_i, ϵ), ei systeemi tulosta mitään, mutta mikäli arvo on muotoa (q_i, a), lisää systeemi tähänastisen tulosteen perään symbolin a. Syöte on sallittu, mikäli syöte saattaa automaatin johonkin sallituista lopputiloista.

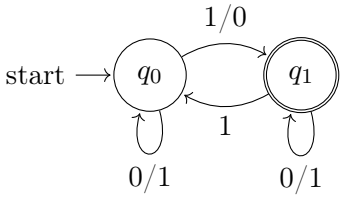
Esimerkki 14. Ohessa yksinkertainen automaatti, joka ei tulosta ulos mitään, ja jonka sallitut syötteet ovat ne merkkijonot joissa on pariton määrä symboleita 1.



Äärellinen automaatti muotoa $(\{q_0, q_1\}, \{0, 1\}, \delta, q_0, \{q_1\})$, missä

$$\delta(q_0, 0) = (q_0, \varepsilon), \quad \delta(q_0, 1) = (q_1, \varepsilon),$$
$$\delta(q_1, 0) = (q_1, \varepsilon), \quad \delta(q_1, 1) = (q_0, \varepsilon),$$

Automaatti voi myös tulostaa kamaa ulos. Seuraava automaatti on kuten yllä, mutta tulostaa ulos symbolin 1 aina kun syötteenä on nolla. Tämä automaatti myöskin hyväksyy ne syötteet, joissa on pariton määrä ykkösiä ja "laskee" syötteen nollien määrän tulostamalla nollien lukumäärän verran ykkösiä.



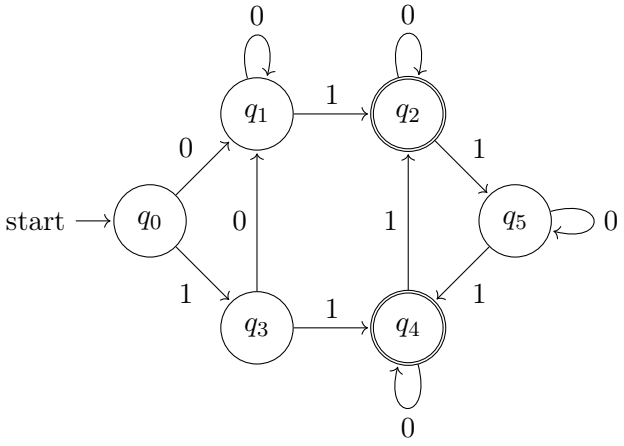
Tämä automaatti on muotoa $(\{q_0, q_1\}, \{0, 1\}, \delta, q_0, \{q_1\})$, missä

$$\delta(q_0, 0) = (q_0, 1), \quad \delta(q_0, 1) = (q_1, \varepsilon),$$
$$\delta(q_1, 0) = (q_1, 1), \quad \delta(q_1, 1) = (q_0, \varepsilon),$$

Esimerkki 15. Seuraava esimerkki on astetta monimutkaisempi äärellinen automaatti muotoa

$$(\{q_0, q_1, q_2, q_3, q_4, q_5\}, \{0, 1\}, \delta, q_0, \{q_2, q_4\}),$$

jonka siirtymäfunktion määrittelyn jätämme oheisen kuvan varaan.



Äärellinen automaatti on erittäin yksinkertainen esimerkki Turingin koneesta, mutta se ei ikinä voi olla Turing-täydellinen. Turing-täydelliset Turingin koneet eivät kuitenkaan ole (määrittelyltään) hirveästi monimutkaisempia kuin äärelliste automaatit; suurin yksittäinen ero on että yleinen Turingin kone voi lukea ja kirjoittaa syötteitä rajoittamattomaan muistinauhaan. Täten, vaikka emme käykään läpi Turingin koneiden formalismia

tämän enempää, niin **Mystisk Boxeja käsitellessä kannattaa pitää äärellinen automaatti mielessä. Monimutkaisintakin tietokonetta voi simuloida Turingin koneella joka näyttää päälle aika samanlaiselta kuin äärellinen automaatti – loppupeleissä kaikki Mystisk Boximme vain siirtyvät syötteen mukaan tilasta toisee ja mahdollisesti tuottavat välillä symboleita.**

Jatkossa oletamme, että Mystisk Boxeilla on universaaliominaisuus – jos U ja T ovat kaksi Mystisk Boxia, niin on olemassa merkkijono π siten, että kaikilla merkkijonoilla S, $U(S) = T(\pi S)$. Kaikilla Mystisk Boxeilla voi siis simuloida kaikkia Mystisk Boxeja.

Määritelmä 3.1.4. Olkoon annettuna aakkosto $\mathcal{A}$ sekä Mystisk Box (Turingin kone) U. Merkkijonon $S \in \mathcal{M}_{\mathcal{A}}$ *Kolmogorov-kompleksisuus* on

$$K_U(S) = \min\{|x| : U(x) = S\}.$$

mikäli U on selvä kontekstista, merkitään $K(\cdot) := K_U(\cdot)$.

Huomataan heti alkuun, että

1. Mystisk Box -universaaliominaisuuden nojalla, mikäli U ja T ovat kumpikin Mystisk Boxeja, niin $K_U(S) = K_T(S) + C$ kaikilla S, missä C riippuu vain Mystisk Boxeista U ja T.

2. Edellisestä edelleen seuraa, että aina $K_U(S) \leq |S| + c_U$.

3.2 Kompleksisuuden ominaisuuksia

Jatkossa oletamme, että taustalla on joku yhteisesti sovittu aakkosto sekä jokin Mystisk Box.

Määritelmä 3.2.1. Merkkijono S on *c-kompressoituva*, $c > 0$, mikäli

$$K(S) \leq |S| - c.$$

Lemma 3.2.2. *Kun aakkostossa on vähintään kaksi symbolia, niin kaikilla $N \in \mathbb{N}$ on olemassa $S \in \mathcal{M}_{\mathcal{A}}^N$ joka ei ole c-kompressoituva millään $c > 0$.*

Todistus. Huomataan, että mikäli aakkoston koko on $k = |\mathcal{A}|$, niin mahdollisia pituuden N mittaisia merkkijonoja on k^N kappaletta.

Toisaalta mahdollisia enintään $N - 1$ mittaisia merkkijonoja on

$$S_{N-1} = k^0 + k^1 + \ldots + k^{n-1} = \frac{k^n - 1}{k - 1}$$

kappaletta. Kun $k \geq 2$, niin $S_{N-1} = k^{N-1} - 1$, eli mahdollisia $N - 1$ pituisia kuvailuja on vähemmän kuin kuvailtavia kohteita. Täten on oltava ainakin yksi N pituinen merkkijono jonka ilmaisuun tarvitaan vähintään N merkkiä.

(Huomaa, että kun aakkoston koko kasvaa, niin edes teoriassa kompressoitavien merkkjonojen määrä vähenee. Myöskin, tämä on karkea arvio koska yleensä kaikki mahdolliset enintään $N-1$ pituiset merkkijonot eivät ole sallittuja syötteitä.) □

Se, että kaikkia merkkijonoja ei voi kompressoida, ei usein arkitilanteissa haittaa, sillä useinmiten kohtaamamme merkkijonot ovat hyvin spesifejä, ja niillä on ominaisuuksia jotka mahdollistavat kompressoinnin.

"Satunnainen" kuva netistä.

Oikea satunnaiskuva.

Lemma 3.2.3. *Jos $x \in \mathcal{M}$ on merkkijono, niin.*

$$K(xx) \leq K(x) + c.$$

Todistuksen idea. Todistus perustuu siihen, että jos meillä on Mystisk Box joka tuottaa syötteellä p merkkijonon x, niin voimme lisätä syötteen alkuun äärellisen mittaisen pätkän π joka "kertoo" boxille että pian syntyvä merkkijono tulee tuplata. □

Se, miksi seuraavassa lemmassa termin $K(x)$ eteen ilmestyy vakio, saattaa tuntua yllättävältä. Tilannetta voi lähestyä ajattelemalla, jos Mystisk Boxin syöte pD koostuu kahdesta osasta, joista alkupää p on 'ohjelmakoodi' ja loppupää D on ohjelmakoodin data, niin kahden peräkkäisen tällaisen syötteen kanssa voi olla hankala erottaa milloin ensimmäisen syötteen data loppuu ja toisen syötteen 'ohjelmakoodi' alkaa.

Lemma 3.2.4. *Jos $x, y \in \mathcal{M}$ ovat merkkijonoja, niin.*

$$K(xy) \leq 2K(x) + K(y) + c$$

Todistuksen idea. Tutkitaan tilannetta, jossa aakkosto on binääriaakkosto.

Jos meillä on syötteet p ja q s.e. $U(p) = x$ ja $U(q) = y$, niin muokkaamme syötteen $p = p_1 p_2 \cdots p_k$ muotoon $\tilde{p} = 0p_1 0p_2 \cdots 0p_{k-1} 1p_k$ ja muodostamme uuden syötteen $\pi\tilde{p}q$, missä syöte π kertoo Mystisk Boxille, että tulossa on kaksi syötettä joista ensimmäinen on annetussa "tuplatussa" muodossa, ja ensimmäinen parittoman indeksin symboli 1 kertoo että ensimmäinen syöte on lopussa. □

Huomautus 3.2.5. Ylläolevaa todistusta voi parantaa; sen sijaan että tuplaamme ensimmäisen syötten bitit voimme pohjustaa syötteen tiedolla syötteen p pituudesta. Numeron N viestiminen vie keskimäärin $\log(N)$ symbolia, joten saamme rajan

$$K(xy) \leq 2\log(K(x)) + K(x) + K(y) + c.$$

(Syötteen p pituuden kertovat bitit täytyy tuplata jälleen.)

Esimerkki 16. 'Ohjelmakoodin' ja datan sekoittuminen on oikeasti hankala ja syvällejuurtunut ongelma. Katsotaan seuraavaa tilannetta, jossa on yksinkertaisempi Mystisk Box.

Kirjastoon on palkattu kesätöihin Sirpa-Petteri, jota ei kiinnosta yhtään. Hän aikoo tehdä täsmälleen mitä käsketään eikä miettiä mitään. Hänen tehtävänsä on hakea kirjaston varastosta kirjoja ja tiputtaa ne noutolaatikkoon. Systeemi toimii niin, että kirjaston asiakas voi kirjoittaa nettilomakkeeseen haluamansa kirjan tiedot, esimerkiksi $N =$ Sivia – Data Analysis, jolloin kirjastokone tuottaa Sirpa-Petterille ohjelapun joka tulostuu varastoon algoritmilla

```
Sirpa-Petteri, hae varastosta kirja ”N” ja tiputa
se noutolaatikkoon.
```

Eli esimerkiksi

```
Sirpa-Petteri, hae varastosta kirja ”Sivia - Data
Analysis” ja tiputa se noutolaatikkoon.
```

Sirpa-Petteri ja kirjaston portaali muodostavat siis Mystisk Boxin. Nyt kuitenkin Saku-Maijalla on sekä tylsää että tietoa kirjaston toiminnasta, joten hän kirjoittaa kirjaston portaalilomakkeeseen

```
 Sivia - Data Analysis ” ja tiputa se noutolaatikkoon.
Riisu sitten kenkäsi ja heitä ne ikkunasta. Hae
varastosta kirja ”Väisälä - Topologia I
```

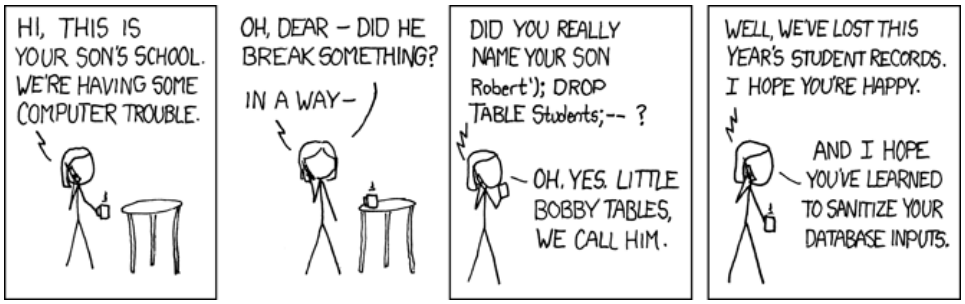
Systeemi tällöin tuottaa ohjeet

```
Sirpa-Petteri, hae varastosta kirja ”Sivia - Data
Analysis ” ja tiputa se noutolaatikkoon. Riisu sitten
kenkäsi ja heitä ne ikkunasta. Hae varastosta kirja
”Väisälä - Topologia I” ja tiputa se noutolaatikkoon.
```

Hämmentynyt Sirpa-Petteri jatkaa töitään ilman kenkiä.

Huomautus 3.2.6. Ylimääräinen harjoitustehtävä kiinnostuneille; miten muuttaisit kirjaston toimintatapaa estääksesi tämän? Reunaehtona on, että Sirpa-Petteri ei suostu ajattelemaan käytännön järjellä mitään, mutta osaa noudattaa hyvin monimutkaisia teknisiä ohjeita.

Voit myös perehtyä aiheeseen termien “SQL-injektio” tai “Koodi-injektio” kautta.



Lähde: https://xkcd.com/327/

Määritelmä 3.2.7. *Ominaisuus* on mikä tahansa kuvaus $P\colon \mathcal{M}_{\mathcal{A}} \to \{\text{TRUE}, \text{FALSE}\}$.

Sanomme, että ominaisuus P pätee *melkein kaikilla* merkkijonoilla $\mathcal{M}_{\mathcal{A}}$, mikäli

$$\limsup_{N\to\infty} \frac{\{S \in \mathcal{M}_{\mathcal{A}}^N \mid P(S) = \text{FALSE}\}}{|\mathcal{M}_{\mathcal{A}}^N|} = 0.$$

(Toisin sanoen niiden merkkijonojen joilla ominaisuus ei päde suhteellinen osuus kaikista merkkijonoista painuu nollaan kun merkkijonojen pituus kasvaa.)

Seuraava lause sanoo, että merkkijonot joilla on jokin harvinainen ominaisuus voidaan aina kompressoida.

Lause 3.2.8. *Jos P on ominaisuus joka pätee melkein kaikilla merkkijonoilla, ja $b > 0$, niin ainoastaan äärellisen moni ei-b-kompressoituva merkkijono ei toteuta ehtoa P.*

Todistuksen idea. Rajoitutaan tarkastelemaan tilannetta, missä aakkostona on binääriaakkosto $\{0, 1\}$. Tässä tilanteessa muistetaan, että luonnollisen luvun n binääriesitys vie $\log_2(n)$ symbolia.

Muodostetaan syötteen $i \in \mathbb{N}$ ottava algoritmi (eli Mystisk Box) T seuraavasti.

1. Lähde käymään läpi kaikkia merkkijonoja kasvavassa järjestyksessä (Eli ensin $\mathcal{M}^0$, sitten $\mathcal{M}^1$, jne.)

2. Testaa kustakin merkkijonosta päteekö sille ominaisuus P; jos ei päde niin talleta se listaan.

3. Kun saat listaasi i:nnen jäsenen, niin tulosta merkkijono ja lopeta.

Ylläoleva algoritmi voidaan kirjoittaa jollakin äärellisellä määrällä merkkejä c. Täten minkä tahansa listalla olevan merkkijonon S_j Kolmogorov-kompleksisuus on enintään

$$K(S_j) \leq c + K(j).$$

Olkoon nyt n niin suuri, että pituutta n olevissa merkkijonoissa ominaisuutta ei toteuttavien merkkijonojen osuus on alle 2^{-b-c-1}. (Tämä ominaisuus pätee sitten myös kaikille $N \geq n$.)

Olkoon edelleen x pituutta n oleva merkkijono jolla ei ole ominaisuutta P, ja merkitään sen indeksiä yllä mainitussa algoritmin T muodostamassa listassa i_x. (Tällainen merkkijono x löytyy indeksin n valinnan nojalla.) Enintään pituutta n olevia merkkijonoja on $1 + 2 + \ldots + 2^n = 2^{n+1} - 1$ kappaletta, joten

$$i_x \leq \frac{2^{n+1} - 1}{2^{b+c+1}} \leq 2^{n-b-c}.$$

Erityisesti $K(i_x) \leq n - b - c$, ja täten

$$K(x) \leq c + K(i_x) \leq c + n - b - c = n - b.$$

Merkkijono x on siis b-kompressoitavissa.

Nyt siis jokainen tarpeeksi pitkä merkkijono joka ei toteuta ehtoa P on b-kompressoituva, joten ainoastaan äärellisen moni merkkijono joka ei toteuta ehtoa P on ei-b-kompressoituva. □

Katsotaan seuraavaksi vielä yhtä näppärää formaalia systeemiä.

Määritelmä 3.2.9. *Lindenmayerin systeemi* (eli *L-systeemi*) on kolmikko (V, ω, P), missä V on aakkosto (jälleen siis kokoelma symboleja), $\omega \in \mathcal{M}_V$ on *aksiooma* ja P on kokoelma *sääntöjä*; jokainen sääntö on muotoa $a \to S$, missä $a \in V$ ja $S \in \mathcal{M}_V$. Toisin sanoen P on kuvaus joltakin aakkoston osajoukolta kaikkien sanojen joukkoon. Kuvauksen P määrittelyjoukon symboleita kutsutaan *muuttujiksi*.

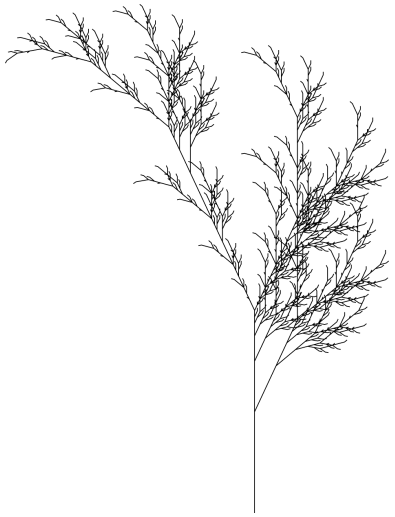
L-systeemin *laajentaminen* askeleella tarkoittaa prosessia, jossa muodostetaan uusi merkkijono korvaamalla aiemmasta merkkijonosta jokainen muuttuja niitä vastaavan säännön antamalla merkkijonolla. Useammalla askeleella laajentaminen tarkoittaa, että laajennetaan systeemiä askeleella, korvataan aksiooma syntyneellä merkkijonolla, ja toistetaan haluttu määrä kertoja.

Esimerkki 17. Otetaan L-systeemi $(\{A, B, +\}, A+, \{A \to BB, B \to +A\})$. Muuttujia ovat nyt A ja B. Laajennokset näyttävät nyt seuraavilta:

$$\begin{aligned}
&\text{askel } 0 : A+ \\
&\text{askel } 1 : BB+ \\
&\text{askel } 2 : +A + A+ \\
&\text{askel } 3 : +BB + BB+
\end{aligned}$$

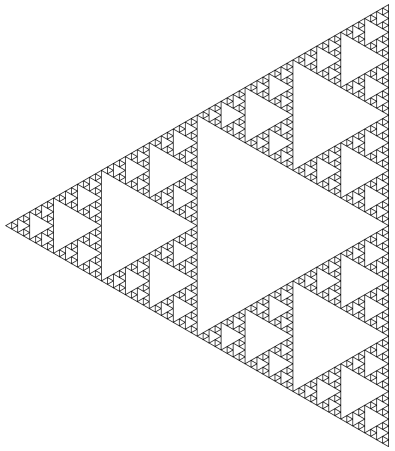
L-systeemeille on omat standardoidut piirto-ohjeensa (esimerkiksi + tarkoittaa "käänny oikealle" ja useimmat aakkoset tarkoittavat "piirrä eteenpäin" tai "älä tee mitään"). Nämä standardoidut piirto-ohjelmat löytyvät esimerkiksi inkscapesta, ja niiden avulla voi piirtää luontevasti fraktaalimaisia asioita.

Erityisesti Lindemayerin systeemeillä voi tuottaa monimutkaisen näköisiä (luontomaisia?) asioita joilla on kuitenkin matala Kolmogorov-kompleksisuus kun ajatellaan L-systeemiä Mystisk Boxina.



Muuttujat : F, X
Muut aakkoset : $+, -, [,]$
Aksiooma : $F - X - X$
Säännöt :
$(X \to F + [[X] - X] - F[-FX] + X)$,
$(F \to FF)$
Kulma : $25°$
Askeleet : 6

Kasvimainen systeemi.



Muuttujat : F, A
Muut aakkoset : $+, -$
Aksiooma : $F - A - A$
Säännöt :
$(F = F - A + F + A - F)$
$(A = AA)$ Kulma : $120°$
Askeleet : 6

Vähemmän kasvimainen systeemi.

3.3 Syventävä laajennos

Perehdytään hieman lisää Turing-Täydellisten koneiden (eli mystisk boxien) rajoitteisiin.

3.3.1 Universaalisuusominaisuus

Tärkeä käsite meille on aiemmin mainittu universaalisuusominaisuus; on olemassa universaali Mystisk Box U siten, että kaikilla Mystisk Boxeilla tai algoritmeilla T on olemassa merkkijono π_U^T siten, että

$$T(x) = U(\pi_U^T x)$$

kaikilla merkkijonoilla x. Kun taustalla oleva universaali Mystisk Box on kontekstista selvä, merkitsemme $\langle T \rangle := \pi_U^T$, jolloin siis kaikilla merkkijonoilla pätee

$$T(x) = U(\langle T \rangle x).$$

Kutsumme usein merkkijonon $\langle T \rangle x$ ensimmäistä osaa ohjelma(koodi)ksi ja toista osaa dataksi. Kiinnostavaa on kuitenkin huomata, että näiden kahden osan ero alkaa vahvasti hämärtyä; kummatkin ovat lopulta vain merkkijonoja. Erityisesti voimme muodostaa algoritmeja, jotka ottavat syötteenä toisten algoritmien ohjelmakoodeja. Tällä on kauaskantoisia seurauksia.

3.3.2 Pysähtymisongelma

Oletamme tässä kappaleessa, että meillä on taustalla jokin universaali Mystisk Box U, jonka 'sisällä' muut algoritmit ajetaan. Oletamme myöskin, että kun syötämme Mystisk Boxiin U ohjelman ja syötteen muodossa $\langle T \rangle x$, niin olemme jollakin keinolla (esimerkiksi bittejä tuplaamalla) pitäneet huolta että ohjelman $\langle T \rangle$ ja datan x raja on selvillä.

Pysähtymisongelma (halting problem) kysyy, että onko olemassa algoritmia joka kertoo jokaisesta Mystisk Boxista, että pysähtyykö se äärellisessä ajassa. Pysähtymisongelman ratkaisu on yksi teoreettisen tietojenkäsittelytieteen sekä matemaattisen logiikan perustuloksista, ja sanoo että tällaista algoritmia ei voida löytää.

Oletetaan nyt, että meillä olisi Mystisk Box H joka ratkaisee pysähtymisongelman, eli että kaikilla Mystisk Boxeilla M pätee, että

$$H(\langle M \rangle x) = \begin{cases} 1, & \text{kun } M \text{ pysähtyy syötteellä } x, \\ 0, & \text{kun } M \text{ ei pysähdy syötteellä } x. \end{cases}$$

Määritellään nyt algoritmi C asettamalla

$$C(\langle M \rangle) = \begin{cases} \texttt{Tulosta 1 ja lopeta}, & \text{jos } H(\langle M \rangle \langle M \rangle) = 0 \\ \texttt{loop}, & \text{jos } H(\langle M \rangle \langle M \rangle) = 1. \end{cases}$$

Tässä "`loop`" tarkoittaa, että algoritmi C aloittaa tahallisesti päättymättömän loopin; tällainen on mahdollista sillä osaamme tuottaa tällaisen Python3:lla: `while 1 < 2: print("hei")` ja täten universaalisuusominaisuuden nojalla voimme kopioida tällaisen loopin Mystisk Box U:n sisälle.

Katsotaan nyt mitä käy algoritmille C, kun sille annetaan syötteenä sen oma ohjelmakoodi: $C(\langle C\rangle)$. Mikäli algoritmi pysähtyy tällä syötteellä, eli $H(\langle C\rangle\langle C\rangle) = 1$, niin algoritmi ajautuu looppiin eikä pysähdy. Jos taas algoritmi ei pysähdy tällä syötteellä, eli $H(\langle C\rangle\langle C\rangle) = 0$, niin algoritmi tulostaa ykkösen ja pysähtyy. Kumpikaan vaihtoehto ei siis ole mahdollinen, ja ollaan ajauduttu ristiriitaan. Täten halutunlaista algoritmia H ei voi olla olemassa.

Kolmogorov-kompleksisuuden laskemattomuus

Oletetaan, että meillä on algoritmi T siten, että kaikilla merkkijonoilla S, $T(S) = K(S)$. Määritetään seuraavaksi algoritmi M, jolle annetaan syötteeksi luonnollinen luku n, seuraavasti.

1. Lähde käymään läpi kaikkia merkkijonoja kasvavassa järjestyksessä (Eli ensin $\mathcal{M}^0$, sitten $\mathcal{M}^1$, jne.)

2. Jokaisen merkkijonon S kohdalla laske sen Kolmogorov-kompleksisuus algoritmilla T.

3. Kun löydät ensimmäisen merkkijonon jolla $K(S) \geq n$, tulosta S ja lopeta.

Merkitään löytynyttä merkkijonoa S_n. Huomaa, että koska kaikilla N on olemassa ainakin yksi merkkijono $S \in \mathcal{M}^N$ joka ei ole kompressoituva, ja toisaalta $K(S) \leq |S| + c$, niin algoritmi M aina pysähtyy ja tuotaa jonkin merkkijonon.

Huomataan seuraavaksi, että merkkijono $\langle T\rangle n$ kuvailee merkkijonon S_n. Nyt

$$K(\langle T\rangle n) \leq 2K(\langle T\rangle) + K(n) \leq 2|\langle T\rangle| + c_1 + \log(n) + c_2 = C + \log(n).$$

Valitsemalla n niin suureksi, että $n\langle C + \log(n)$, huomataan että

$$n \leq K(S) \leq K(\langle T\rangle n) \leq C + \log(n) < n.$$

Tämä on ristiriita, joten toivotunlaista algoritmia ei voi olla olemassa.

3.3.3 Gödelin epätäydellisyyslause

Kriittinen ominaisuus yllä on se, että universaaliominaisuuden nojalla Mystisk Boxit sekä niihen syötettävä data ovat jossain määrin samassa asemassa. Vastaavanlainen ominaisuus on käsillä matemaattisessa logiikassa, jossa

Gödelin epätäydellisyyslause kertoo, että aina on olemassa matemaattisia lauseita joita ei voi todistaa oikeaksi tai vääräksi.

Eräs tärkeä askel Gödelin epätäydellisyyslauseen todistamisessa on huomata, että loogiset väitteet ja todistukset voidaan koodata yksikäsitteisiksi luonnollisiksi numeroiksi; Gödel-numeroinnissa jokaiselle loogiselle symbolille s annetaan lukuarvo $n_s \in \mathbb{N}$, ja looginen ilmaisu $s_1 s_2 s_3 \cdots s_k$ muunnetaan numeroksi $p_1^{s_1} \cdot p_2^{s_2} \cdot \ldots \cdot p_k^{s_k}$, missä p_j tarkoittaa alkulukua numero j. Tämä numerointi kiinnittää jokaiseen kaavaan, todistukseen ja muuhun loogisilla symboleilla esitettyyn väitteeseen yksikäsitteisen luonnollisen luvun, jonka jälkeen **ero luonnollisten lukujen ja luonnollisia lukuja koskevien väitteiden välillä hämärtyy – aivan kuten Mystisk Boxiemme kanssa.**

3.3.4 Chaitinin Omega ja muita asioita

Maininnan tasolle luennoilla jäi muutama pidemmälle menevä käsite; Busy Beaver -funktio, Chaitinin Omega -vakio sekä muutama erikoisempi esimerkki Turing-Täydellisistä systeemeistä (FRACTRAN ja Rule 110). Käsitteisiin voi perehtyä halutessaan vaikka Wikipedista, Chaitinin omega on lyhyesti sanottuna maksimaalisen ei-kompressoitavissa oleva reaaliluku ja Busy Beaver -funktio kasvaa niin nopeasti ettei sitä voi laskea millään algoritmilla.

Luku 4

Koodausteoriaa ilman kohinaa

Tässä osiossa siirrytään koodausteoriaan, eli keskitytään informaatiosisällön tutkimisesta siihen, miten informaatiota parhaiten siirrettäisiin. Aihe yhdistää aiempia Entropian Kolmogorov-kompleksisuuden ideoita.

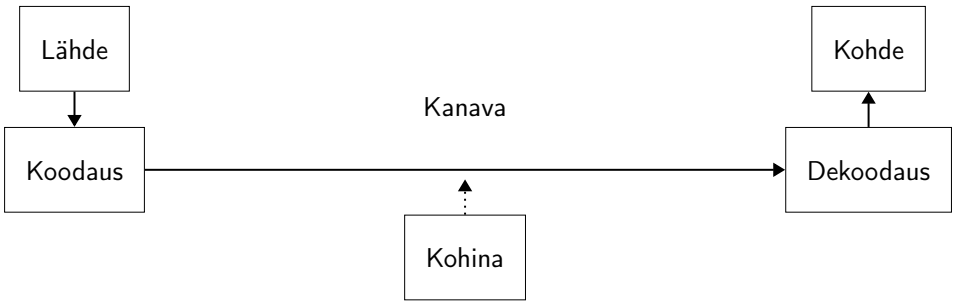
CHARACTER	*MORSE CODE*	*TELEPHONY*	*PHONIC (PRONUNCIATION)*
A	• —	Alfa	(AL-FAH)
B	— • • •	Bravo	(BRAH-VOH)
C	— • — •	Charlie	(CHAR-LEE) or (SHAR-LEE)
D	— • •	Delta	(DELL-TAH)
E	•	Echo	(ECK-OH)
F	• • — •	Foxtrot	(FOKS-TROT)
G	— — •	Golf	(GOLF)
H	• • • •	Hotel	(HOH-TEL)
I	• •	India	(IN-DEE-AH)
J	• — — —	Juliett	(JEW-LEE-ETT)
K	— • —	Kilo	(KEY-LOH)
L	• — • •	Lima	(LEE-MAH)
M	— —	Mike	(MIKE)
N	— •	November	(NO-VEM-BER)
O	— — —	Oscar	(OSS-CAH)
P	• — — •	Papa	(PAH-PAH)
Q	— — • —	Quebec	(KEH-BECK)
R	• — •	Romeo	(ROW-ME-OH)
S	• • •	Sierra	(SEE-AIR-RAH)
T	—	Tango	(TANG-GO)
U	• • —	Uniform	(YOU-NEE-FORM) or (OO-NEE-FORM)
V	• • • —	Victor	(VIK-TAH)
W	• — —	Whiskey	(WISS-KEY)
X	— • • —	Xray	(ECKS-RAY)
Y	— • — —	Yankee	(YANG-KEY)
Z	— — • •	Zulu	(ZOO-LOO)
1	• — — — —	One	(WUN)
2	• • — — —	Two	(TOO)
3	• • • — —	Three	(TREE)
4	• • • • —	Four	(FOW-ER)
5	• • • • •	Five	(FIFE)
6	— • • • •	Six	(SIX)
7	— — • • •	Seven	(SEV-EN)
8	— — — • •	Eight	(AIT)
9	— — — — •	Nine	(NIN-ER)
0	— — — — —	Zero	(ZEE-RO)

Morse- ja radioaakkoset.
Lähde: Wikimedia commons.

Usein tiedonsiirrossa joudutaan viestimään viestintäkanavalla, joka ei ole täysin luotettava, vaan jossa viestiin voi syntyä ennakoimattomia virheitä. Virheistä aiheutuvia ongelmia voi vähentää koodaamalla viestit johonkin muotoon josta virheitä voi huomata. Tämä kuitenkin pidentää lähetettävää viestiä, eli yleensä hidastaa viestintänopeutta. Usein joudutaankin tasapainoilemaan viestintänopeuden ja -varmuuden välillä.
Vieressä olevassa kuvassa on rinnakkain sekä Morseaakkoset että radioaakkoset. Huomionarvoista on, että Morseaakkosissa usein esiintyvillä kirjaimilla, kuten A, E ja T, on lyhyemmät koodit kuin harvinaisemmilla kirjaimilla, esimerkiksi Q ja Z. Koska yleisemmillä kirjaimilla on lyhyemmät koodit kuin harvinaisemmilla, voi Morsen aakkosilla viestiä keskimäärin nopeammin kuin koodilla joissa kaikilla kirjaimilla on yhtä pitkät ilmaisut. Toisaalta radioaakkoset ovat poikkeuksetta pidemmät sanoa kuin pelkkä vastaava kirjain – tämä hidastaa viestin välittämistä puhekanavalla mutta parantaa virheensietokykyä.

4.1 Perusosa

Tässä osiossa keskitytään tilanteeseen, jossa kohinaa ei ole tai sitä ei oteta huomioon, eli oletamme että kaikki viestit siirtyvät täydellisen oikein vastaanottajalle. Perusasetelmassa meillä on **lähde**, **koodaus**, **kanava**, **dekoodaus** ja **kohde**. Meidän asetelmassamme lähde- ja kohdeviestit tuottavat ja vastaaottavat sanoja muotoa $\mathcal{M}_{\mathcal{A}}$, koodaus muokkaa merkkijonoja $\mathcal{M}_{\mathcal{A}}$ merkkijonoiksi $\mathcal{M}_{\mathcal{B}}$, ja dekoodaus muokkaa merkkijonoja $\mathcal{M}_{\mathcal{B}}$ merkkijonoiksi $\mathcal{M}_{\mathcal{A}}$.



Havainnekuva tilanteesta.

Rajoitumme tällä kurssilla malliin, jossa lähde tuottaa symboleita $\mathcal{M}_{\mathcal{A}}$ jonkin TN-jakauman antamana. Tutkimme siis n.s. symbolikoodeja, mutta muitakin on.

Viestintäyhteyden parantamista voi lähteä tekemään esimerkiksi suunnittelemalla lähteen ja kohteen välillä kulkevia viestejä (viestikuri) parantamalla kanavan ominaisuuksia (insinööritiedettä) tai suunnittelemalla nokkelampia koodaus- ja dekoodausmentelmiä (koodausteoria). Tällä kurssilla keskitytään koodausteoriaan.

Esimerkki 18. Esimerkkejä kommunikaatiokanavista.

1. Morsetus parikaapelia pitkin: Lähteenä tekstiä, koodauksena morse ja kanavana lennätinlinja. (Eli kanavassa ykkösiä ja nollia.)

2. Puhe ääniaalloilla: Lähteenä tekstiä, koodauksena muutos ääniksi ja kanavana ilmassa tapahtuvat painevaihtelut.

3. DNA-viestintä jälkeläisille: Lähteenä genettistä tietoa, koodauksena muunnos DNA:ksi ja kanavana fyysinen siirto.

4. Datan tallennus kovalevylle ja lukeminen myöhemmin: Lähteenä esimerkiksi kissakuva, koodauksena muutos ykkösiksi ja nolliksi ja kanavana aika.

Kohinattomassa tilanteessa keskitymme kysymykseen, että miten voisimme viestiä mahdollisimman tehokkasti kanavalla, jossa lähetysnopeus on rajoitettu. (Esimerkiksi yksi merkki sekunnissa.) Haluamme siis käytännössä käyttää koodausta *kompressoidaksemme viestejä* mahdollisimman paljon. Tähän liittyy muutama varoitus viime luvusta:

1. 'Useimpia' asioita ei voi kompressoida.

2. Jos yritetään viestiä mahdollisimman tiiviisti asioita, niin peräkkäisten viestin rajat saattavat sekoittua.

Vastaukseemme ongelmaan yksi on, että meidän ei tarvitsekaan kompressoida kaikkia viestejä, vaan meille riittää että kompressoimme *todennäköisimpiä* viestejä. Ongelmaan kaksi puolestaan vastaamme kahdella eri tavalla, ja tarkastelemmekin seuraavaksi sekä **(I) vakiomittaisia viestejä** että **(II) vaihtelevanpituisia viestejä**.

Merkitsemme tässä kappaleessa TN-jakaumaa taas muodossa

$$X = \{(a_1, \ldots, a_k), (p_1, \ldots, p_k)\}.$$

Merkitsemme jakauman X alkeistapauksien joukkoa Ω_X.

Oletamme, ellei toisin sanota, että käytössä olevalla kanavalla tieto kulkee binääriviesteinä.

4.1.1 Vakiomittaiset viestit

Vakiomittaisten viestien tilanteessa meillä on lähteenä jokin TN-jakauma joka tuottaa meille symboleita. Muokkaamme viesteistä koodin kanavalle, ja haluamme pitää kaikkien koodien pituudet samoina. Huomataan, että binäärikanavalla voidaan pituuden k merkkijonolla viestiä 2^k erilaista asiaa, joten suoraan viestittynä lähteenä toimivan TN-jakauman X koodaamiseen vakiomittaisilla viesteillä tarvitaan viestjä, joiden pituus on $\lceil \log_2(|\Omega_X|) \rceil$.

Jos haluamme ottaa huomioon eri symbolien todennäköisyydet, lähes ainut tapa on karsia lähetettyjen symbolien määrää. Lähdetään siis tutkimaan *häviöllistä pakkausta.*

Esimerkki 19. Olkoon X jakauma $\{(a, b, c, d, e), (\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{15}{128}, \frac{1}{128})\}$.

Vakiomittaisilla binäärikoodeilla jakauman viestimiseen tarvitaan pituuden $\lceil \log_2(5) \rceil = \lceil 2.3 \rceil = 3$ mittaisia merkkijonoja. Kuitenkin huomataan, että jättämällä e pois viesteistä menetämme keskimäärin vähemmän kuin yhden merkin sadasta, ja tarvitsemmekin enää $\lceil \log_2(4) \rceil = 2$ merkkiä per symboli viestintään. Lähetysnopeutemme siis kasvaa huomattavasti sallimalla prosenttiluokan virhe lähetyksessä.

Määritelmä 4.1.1. Olkoon $X = ((a_1, \ldots, a_k), (p_1, \ldots, p_k))$ todennäköisyysjakauma ja $\delta \in (0, 1)$. Nyt jakauman *pienin δ-riittävä osajoukko* S_δ on lukumäärältään pienin alkeistapauksien osajoukko jolle pätee

$$\sum_{x \in S_\delta} P(x) \geq 1 - \delta.$$

(Kokoelma S_δ ei ole yleensä yksikäsitteinen, mieti esimerkiksi tasajakaumaa.)

Edelleen, jakauman X *δ-oleellinen bittisisältö* on

$$H_\delta(X) := \log_2(|S_\delta|).$$

Yllä siis oleellinen bittisisältö (pyöristettynä ylöspäin) kertoo montako symbolia keskimäärin tarvitaan, jos δ-riittävän osajoukon haluaa viestiä vakiopituisilla viesteillä.

Määritelmä 4.1.2. Mikäli X on TN-jakauma ja $N \in \mathbb{N}$, merkitsemme merkillä X^N sitä yhteisjakaumaa jossa on N kappaletta toisistaan riippumatonta satunnaismuuttujaa joiden kaikkien marginaalijakauma on X.

Toisin sanoen, X^N on jakauma, jonka alkeistapaukset ovat sanoja $\mathcal{M}^N_{\Omega_X}$, ja kunkin jonon $a_1 a_2 \cdots a_N$ todennäköisyys on

$$P(a_1 a_2 \cdots a_N) = P(a_1) \cdot P(a_2) \cdot \ldots \cdot P(a_N).$$

Esimerkki 20. Olkoon X jakauma $\{(0, 1), (\frac{1}{10}, \frac{9}{10})\}$.

Suurilla N erilaisia viesti'blokkeja' on $|X|^N$ kappaletta, joten niiden viestimiseen tarvitaan vähintään $\lceil \log_2(|X|^N) \rceil = \lceil N \log_2(|X|) \rceil$ merkkiä. Tällöin voidaan ajatella että keskimäärin yhden viestisymbolin viestimiseen kuluu $\frac{1}{N} \lceil \log_2(|X|^N) \rceil \approx \lceil \log_2(|X|) \rceil$ merkkiä, eli suunnilleen saman verran kuin viestittäessä symboleja "suoraan". Kuitenkin, jakaumassa X^N on enemmän alkeistapauksia, jolloin meillä on suurempi valikoima tehdä erilaisia δ-riittäviä osajoukkoja.

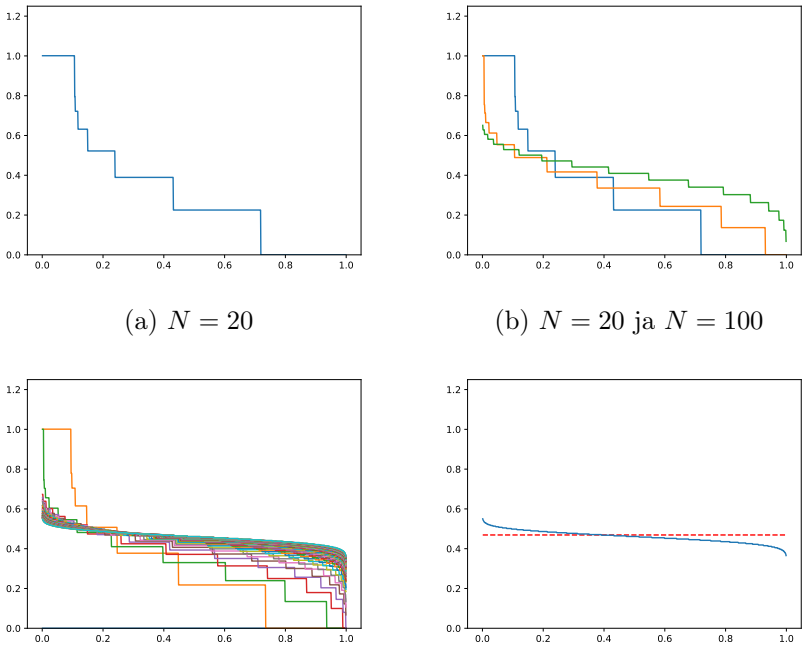
Miltä näyttää $\frac{1}{N} H_\delta(X^N)$ kun N on suuri? Tilannetta on hahmoteltu kuvassa 4.1

Lause 4.1.3 (Shannonin lähdekoodauslause)**.** *Olkoon X todennäköisyysjakauma ja merkitään siitä muodostettua riippumatonta N-kertaista yhteisjakaumaa*

$$X^N := (\underbrace{X, X, \ldots, X}_{n \text{ kpl}}).$$

Tällöin kaikille $\varepsilon > 0$ ja $\delta \in (0, 1)$ löytyy $N_0 \in \mathbb{N}$ siten, että

$$\left| \frac{1}{N} H_\delta(X^N) - H(X) \right| < \varepsilon.$$



(a) $N = 20$ (b) $N = 20$ ja $N = 100$

(c) $N = 20, 40, 60, \ldots, 1000$ (d) $N = 1000$ ja $H(X)$

Kuva 4.1: Funktion $\delta \mapsto \frac{1}{N} H_\delta(X^N)$ hahmottelua kun $X = \{(0,1),(0.1,0.9)\}$. *(Kuvassa on tarkalleen ottaen kuvauksen approksimaatio joka valitsee S_δ-joukot hieman 'röyhkeästi'. Kuvaajan rakenne on kuitenkin aivan oikea.)*

Ottaen huomioon, että $\frac{1}{N}H_\delta(X^N)$ kuvastaa sitä, montako merkkiä meidän tarvitsee keskimäärin lähettää viestiäksemme yhden lähteen merkin, kertoo lause siis että millä tahansa tarkkuudella voidaan lähdettä viestiä suunnilleen sen entropian nopeudella, kun viestiblokkien pituudet vaan valitaan tarpeeksi suuriksi.

4.1.2 Vaihtelevapituiset viestit

Seuraavaksi tarkastellaan tilannetta, jossa ei sallita pakkauksessa häviöitä, mutta jossa viestien pituudet voivat vaihdella. Rajoitutaan tässä kappaleessa binääriseen viestintäkanavaan.

Otetaan käyttöön hieman merkintöjä.

Määritelmä 4.1.4. Olkoon X TN-jakauma. *Koodaus* on funktio $c\colon \Omega_X \to \mathcal{M}_{0,1}$. Merkitään $\ell(a_i) = \ell_i = |c(a_i)|$, ja mikäli $S = a_1a_2\cdots a_k \in \mathcal{M}_{\Omega_X}$, niin merkitsemme

$$C(S) = C(a_1a_2\cdots a_k) := c(a_1)c(a_2)\cdots c(a_k).$$

Asetetaan edelleen koodauksen *odotuspituudeksi*

$$L(c, X) = E[\ell] = \sum_j p(a_i)\ell(a_i).$$

Haluamme löytää koodauksia jotka minimoivat odotuspituuden.

Määritelmä 4.1.5. TN-jakauman koodaus on *optimaalinen*, mikäli ei ole olemassa toista koodausta $\tilde{c}$ s.e. $L(\tilde{c}, X) < L(cX)$.

Esimerkki 21. Olkoon X TN-jakauma

$$\{(a, b, c, d, e), (\frac{1}{2}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8})\}.$$

Muodostetaan koodaukset c ja $\tilde{c}$ seuraavasti:

$$c(x) = \begin{cases} 001, & \text{kun } x = a \\ 1, & \text{kun } x = b \\ 1001001, & \text{kun } x = c \\ 0, & \text{kun } x = d \\ 101, & \text{kun } x = e \end{cases} \quad \tilde{c}(x) = \begin{cases} 0, & \text{kun } x = a \\ 100, & \text{kun } x = b \\ 101, & \text{kun } x = c \\ 110, & \text{kun } x = d \\ 111, & \text{kun } x = e \end{cases}$$

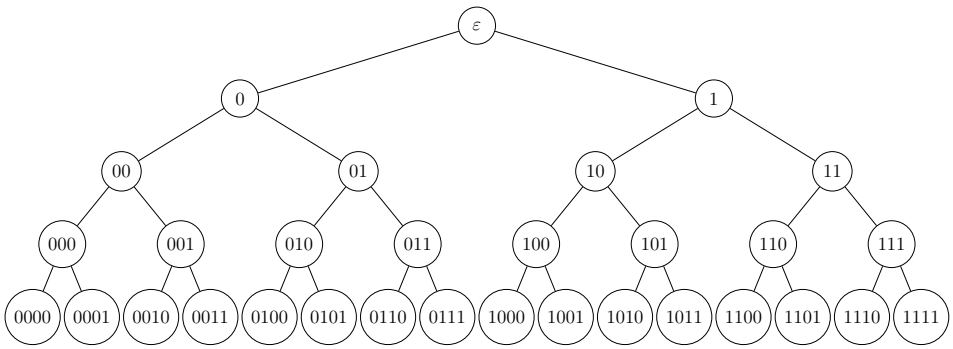
Näille koodeille voidaan laskea odotuspituudet $L(c, X) = 3$ ja $L(\tilde{c}, X) = \frac{7}{8}$.

Määritelmä 4.1.6. Koodaus c on *yksikäsitteisesti purettavissa*, mikäli ei ole olemassa kahta eri merkkijonoa $S, T \in \mathcal{M}_{\Omega_X}$ joille pätee $C(T) = C(Y)$

Koodaus c on *etuliitevapaa*, mikäli ei ole olemassa $a, b \in \Omega_X$ ja $S \in \mathcal{M}_{0,1}$ siten, että $c(a) = c(b)S$.

Esimerkki 22. Edellisen esimerkin koodi c' sekä yksikäsitteisesti purettavissa että etuliitevapaa, kun c puolestaan ei ole kumpaakaan.

Koodin yksikäsitteisen purettavuuden tarkistaminen on epätriviaalia, mutta etuliitevapaus on suoraviivaisempaa. Myöhemmin Kraft-McMillanin epäyhtälö kertoo meille että lähes aina voidaan rajoittua etuliitevapaisiin koodeihin. Etuliitevapaata koodia kannattaa miettiä ajattelemalla nk. binääripuuta, jossa kaikki binäärijonot on koottu puurakenteeseen seuraavan kuvan hengessä:



Koodauksen valitseminen vastaa sitä, että valitsemme symboleille koodeja tällaisesta puusta. *Etuliitevapaan koodauksen valitseminen vastaa sitä, että valitsemme koodeja tällaisesta puusta siten, että mikään koodi ei ole toisen koodan aiempi 'haara'.*

Esimerkki 23. Meidän tilanteessamme paras tapa generoida etuliitevapaita koodeja on n.k. *Huffmanin koodaus.*

Olkoon X TN-jakauma. Tee seuraavasti.

1. Ota kaksi epätodennäköisintä symbolia alkeistapauksien joukosta. Näille symboleille tullaan valitsemaan pisimmät koodit, jotka poikkeavat toisistaan vain viimeisen bitin osalta.

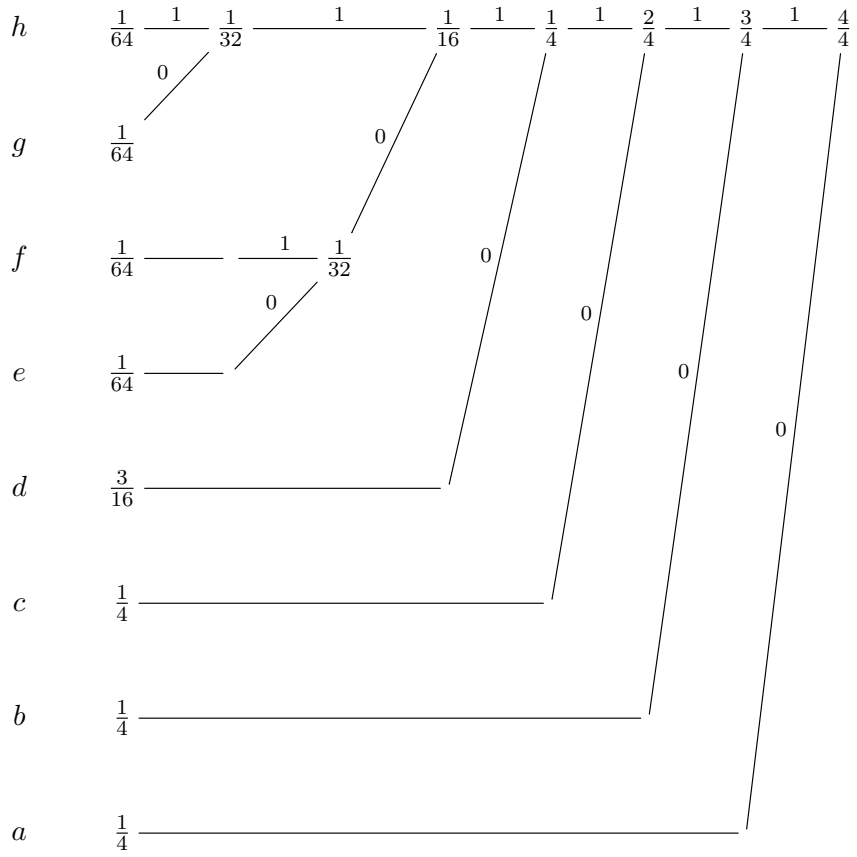
   Valitse toiselle symboli 1 ja toiselle symboli 0.

2. Muodosta uusi TN-jakauma, jossa aiemman kohdan alkeistapauspari yhdistetään yksittäiseksi alkeistapaukseksi ja muut alkeistapaukset pidetään ennallaan. Toista kohtia 1 ja 2 kunnes jäljellä on enää yksi alkeistapaus.

Algoritmin jälkeen jokainen alkuperäinen alkeistapaus kuuluu johonkin jonoon sisäkkäisiä alkeistapauspareja, joihin kuhunkin on liitetty 1 tai 0. Tämä binäärijono on nyt alkuperäisen alkeistapauksen koodi.

Tilanne kannattaa piirtää "puumuotoon", kts. seuraava esimerkki.

Esimerkki 24. Muodostetaan jakauman $(\{a,b,c,d,e,f,g,h\}, \{\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{3}{16}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}\})$ Huffman-koodi.



Koodiksi siis saadaan:

Symboli:	a	b	c	d	e	f	g	h
Koodi:	0	10	110	1110	111100	111101	111110	111111
Pituus:	1	2	3	4	6	6	6	6

Josta edelleen

$$L(c,X) = 1 \cdot \frac{1}{4} + 2 \cdot \frac{1}{4} + 3 \cdot \frac{1}{4} + 4 \cdot \frac{3}{16} + 4\left(6 \cdot \frac{1}{64}\right) \approx 2.625.$$

Esimerkki 25. Aiemman esimerkin 21 koodi $\tilde{c}$ on Huffmanin koodi.

Lause 4.1.7 (Kraft-McMillan)**.** *Olkoon X TN-jakauma.*

1. *Jos c on yksikäsitteisesti purettavissa oleva binäärikoodi, niin tällöin*

$$\sum_{x \in \Omega_X} 2^{-\ell(c(x))} \leq 1.$$

2. *Jos on annettuna pituudet l_i, $i = 1, \ldots, |\Omega_X|$, siten että $\sum_i 2^{-l_i} \leq 1$, niin tällöin on olemassa yksikäsitteisesti purettavissa oleva etuliiteva-paa koodi $\tilde{c}$ siten, että $\ell(c(a_i)) = l_i$ kaikilla $a_i \in \Omega_X$.*

Todistus. Todistetaan syventävässä laajennoksessa. □

Lause 4.1.8. *Olkoon X TN-jakauma ja c yksikäsitteisesti purettavissa oleva binäärikoodi. Tällöin $L(c, X) \geq H(X)$.*

Todistus. Merkitään taas koodisanojen pituuksia $\ell_i := \ell(c(a_i))$ ja asetetaan $z_0 = \sum_j 2^{-\ell_j}$. Nyt kokoelma $q_i := 2^{-\ell_i}/z_0$ antaa uuden todennäköisyysjakauman jakauman X määrittelyjoukolle. Erityisesti nyt $\ell_i = \log_2(1/q_i) - \log_2(z_0)$.

Gibbsin epäyhtälön nojalla

$$\sum_i p_i \log_2(\frac{1}{q_i}) \geq \sum_i p_i \log_2(\frac{1}{p_i}),$$

ja Kraft-MacMillanin epäyhtälön nojalla $z \leq 1$, joten

$$\begin{aligned} L(c, X) &= \sum_i p_i \ell_i \\ &= \sum_i p_i \log_2(1/q_i) - \sum_i p_i \log_2(z_0) \\ &= \sum_i p_i \log_2(1/q_i) - \log_2(z_0) \\ &\geq \sum_i p_i \log_2(1/p_i) - \log_2(z_0) \\ &\geq \sum_i p_i \log_2(1/p_i) = H(X). \end{aligned}$$

□

Huomautus 4.1.9. Huffmanin koodille pätee $L(c, X) < H(X) + 1$.

4.2 Syventävä laajennos

Tässä osiossa todistetaan Shannonin lähdekoodauslause sekä Kraft-McMillaninin epäyhtälö.

4.2.1 Shannonin lähdekoodauslause

Palautetaan mieleen, että TN-jakaumassa (Ω, P) määritellyn reaaliarvoisen satunnaismuuttujan $X\colon \Omega \to \mathbb{R}$ *odotusarvo* on

$$E[X] := \sum_{a\in\Omega} P(a)X(a)$$

ja *varianssi tai keskihajonta* on

$$var^2(X) = E[(X - E[X])^2].$$

Sanomme, että samassa todennäköisyysjakaumassa $(\Omega, \{p_1, \ldots, p_n)$ määritellyt satunnaismuuttujat X ja Y ovat riippumattomat, mikäli yhteisjakaumassa $(\Omega^2, \{p_i p_j \mid i, j = 1, \ldots, n\}$ määritelty satunnaismuuttuja $(a, b) \mapsto (X(a), X(b))$ toteuttaa

$$P(X = a \wedge Y = b) = P(X = a)P(Y = b)$$

kaikilla $a, b \in \Omega$.

Seuraava tulos sanoo, että satunnaismuuttujan on epätodennäköistä saada arvoja jotka ovat kaukana odotusarvosta.

Lause 4.2.1 (Chebyshevin ensimmäinen epäyhtälö)**.** *Olkoon (Ω, P) TN-jakauma ja X reaaliarvoinen epänegatiivinen satunnaismuuttuja $X\colon \Omega \to [0, \infty)$. Tällöin kaikilla $\alpha > 0$ pätee*

$$P(X \geq \alpha) \leq \frac{E[X]}{\alpha}.$$

Todistus. Merkitään $\Omega_\alpha = \{a \in \Omega \mid X(a) \geq \alpha$ ja huomataan, että

$$\begin{aligned} P(X \geq \alpha) &= \sum_{a\in\Omega_\alpha} P(a) = \sum_{a\in\Omega_\alpha} P(a)\frac{\alpha}{\alpha} \\ &\leq \sum_{a\in\Omega_\alpha} P(a)\frac{X(a)}{\alpha} \leq \frac{1}{\alpha}\sum_{a\in\Omega_\alpha} P(a)X(a) \\ &\leq \frac{1}{\alpha}\sum_{a\in\Omega} P(a)X(a) = \frac{E[X]}{\alpha} \end{aligned}$$

□

Lause 4.2.2 (Chebyshevin toinen epäyhtälö)**.** *Olkoon* (Ω, P) *TN-jakauma ja* X *reaaliarvoinen satunnaismuuttuja* $X\colon \Omega \to \mathbb{R}$*. Tällöin kaikilla* $\alpha > 0$ *pätee*

$$P((X - E[X])^2 \geq \alpha) \leq \frac{var^2(X)}{\alpha}.$$

Todistus. Seuraa Chebyshevin ensimmäisestä epäyhtälöstä soveltamalla sitä epänegatiiviseen satunnaismuuttujaan $(X - E[X])^2$. □

Lause 4.2.3 (Heikko suurten lukujen laki)**.** *Olkoon* (Ω, P) *TN-jakauma ja olkoot* $h_1, \ldots, h_n\colon \Omega \to \mathbb{R}$ *riippumattomia satunnaismuuttujia, joilla kaikilla on sama odotusarvo* $\overline{h}$ *ja varianssi* σ_h^2*. Muodostetaan uusi satunnaismuuttuja* $X\colon \Omega^n \to \mathbb{R}$ *asettamalla* $X(a) = \frac{1}{n}\sum_j h_j(a)$*. Tällöin*

$$P((X - \overline{h})^2 \geq \alpha) \leq \frac{var^2(X)}{\alpha n}.$$

Todistuksen idea. Näytä, että $E[X] = \overline{h}$ ja $var^2(X) = \sigma_h^2$, ja sovella Chebyshevin toista epäyhtälöä. □

Tyypilliset merkkijonot

Lähdetään käymään läpi muutamia perusideoita joita tarvitaan Shannonin lähdekoodauslauseen todistuksessa. Rajoiteutaan tarkastelemaan tilannetta, jossa lähteenä on edelleen TN-jakauma $((0,1),(p_0,p_1))$, missä $p_0 = \frac{1}{10}, p_1 = \frac{9}{10}$. Todistus yleisessä tilanteess näyttää lähes samalta. Merkitään edelleen $X\colon \Omega \to \Omega$, $X(0) = 0, X(1) = 1$ satunnaismuuttujaa, joka kuvaa vain alkeistapaukset itselleen – merkitsemme usein jatkossa TN-jakauman entropiaa $H(X)$ ja sanomme että tutkimme X^N-jakauman alkioita, vaikka teknisesti ottaen X on satunnaismuuttuja eikä jakauma. Koitetaan kestää.

Muodostetaan taas N-kertainen jakauma (Ω^N, P), missä $P(a_1, \ldots, a_N) = P_{X_1}(a_1) \ldots P_{X_N}(a_N)$; merkitsemme selvyyden vuoksi jakauman (Ω^N, P) marginaalijakaumien todennäköisyysfunktioita P_{X_j} tai P_X – koska jakauma rakennettiin riippumattomana tulojakaumana ovat kaikki marginaalijakaumat samoja, muotoa $P_X(0) = p_0$, $P_X(1) = p_1$. Puhumme jatkossa ajoittain "jakaumasta X^N", vaikka teknisesti ottaen kyseessä on satunnaismuuttuja; notaatio on vaan vähän kevyempää näin.

Miltä näyttää "tyypillinen" jakauman X^N alkio S? Kun N on hyvin iso, on luontevaa odottaa, että merkkijonossa on noin Np_0 nollaa ja Np_1 ykköstä. Tällöin voidaan laskea, että

$$\begin{aligned} h(S) &= -\log_2(P(S)) = -\log_2(P(a_1 a_2 \cdots a_N)) \\ &= -\log_2(P_X(a_1)P_X(a_2)\cdots P_X(a_N)) \approx -\log_2(p_0^{Np_0} \cdot p_1^{Np_1}) \\ &= N(-p_0\log_2(p_0) - p_1\log_2(p_1) = NH(X) = H(X^N) \end{aligned}$$

Tyypillisellä merkkijonolla on siis tyypillinen informaatiosisältö, sillä entropia on määritelmänsä nojalla informaatiosisällön odotusarvo. Huomaa myös, että koska tyypilliselle merkkijonolle pätee $h(S) = -\log_2(P(S)) \approx NH(X)$, niin erityisesti saadaan $P(S) \approx 2^{-NH(X)}$.

Määritelmä 4.2.4. Olkoon $\beta > 0$. Määritellään *βN-tyypillisten merkkijonojen kokoelmaksi*

$$T_{N\beta} := \left\{ S \in \mathcal{M}^N_{\{0,1\}} : \left| \frac{1}{N} \log_2 \left(\frac{1}{P(S)} \right) - H(X) \right| < \beta \right\}.$$

Lemma 4.2.5. *Kaikilla $\beta > 0$,*

$$P(S \in T_{N\beta}) \geq 1 - \frac{\sigma^2}{\beta^2 N},$$

missä σ^2 on satunnaismuuttujan X varianssi.

Todistus. Määritetään satunnaismuuttujat

$$h_j \colon \Omega^N \to \mathbb{R}, \quad h_j(a_1 a_2 \cdots a_N) = -\log_2(P_X(a_j)).$$

Huomataan, että

$$g(S) := \frac{1}{N} \sum_j h_j(S) = -\frac{1}{N} \sum_j \log_2(P_X(a_j)) = \frac{1}{N} \log_2 \left(\frac{1}{P(S)} \right),$$

joten heikon suurten lukujen lain nojalla

$$P(|g(S) - H(X)|^2 > \alpha) \leq \frac{\sigma^2}{\alpha N},$$

joten erityisesti

$$P(S \in T_{N\beta} \geq 1 - \frac{\sigma^2}{\beta^2 N}.$$

□

Lemma 4.2.6. *Kaikilla $S \in T_{\beta N}$ pätee*

$$2^{-N(H+\beta)} < P(S) < 2^{-N(H-\beta)},$$

missä $H := H(X)$.

Todistus. Huomataan, että jos $S \in T_{N\beta}$, niin

$$\begin{aligned}
&\left| \frac{1}{N} \log_2 \left(\frac{1}{P(S)} \right) - H(X) \right| < \beta \\
\Leftrightarrow &-\beta < \frac{1}{N} \log_2 \left(\frac{1}{P(S)} \right) - H(X) < \beta \\
\Leftrightarrow &N(H(X) - \beta) < \log_2 \left(\frac{1}{P(S)} \right) < N(H(X) + \beta) \\
\Leftrightarrow &2^{-N(H(X)-\beta)} > P(S) > 2^{-N(H(X)+\beta)}.
\end{aligned}$$

□

Shannonin lähdekoodauslause

Lause 4.2.7 (Shannonin lähdekoodauslause)**.** *Olkoon X todennäköisyysjakauma ja merkitään siitä muodostettua riippumatonta N-kertaista yhteisjakaumaa*

$$X^N := (\underbrace{X, X, \ldots, X}_{N \text{ kpl}}).$$

Tällöin kaikille $\varepsilon > 0$ ja $\delta \in (0,1)$ löytyy $N_0 \in \mathbb{N}$ siten, että

$$\left| \frac{1}{N} H_\delta(X^N) - H(X) \right| < \varepsilon.$$

Todistus. Kiinnitetään $\varepsilon > 0$ ja $\delta \in (0,1)$. Todistetaan väite kahdessa osassa.

Osa 1: $\frac{1}{N} H_\delta(X^N) \leq H(X) + \varepsilon$

Asetetaan $\beta = \varepsilon$ ja tutkitaan tyypillisiä merkkijonoja. Kun N on tarpeeksi suuri, niin $\frac{\sigma^2}{\beta^2 N} \leq \delta$, joten erityisesti tarpeeksi isoilla N pätee

$$P(S \in T_{\beta N}) \geq 1 - \delta. \tag{4.2.1}$$

Koska δ-riittävä osajoukko on määritelmänsä nojalla pienin osajoukko, jolla (4.2.1) pätee, joten välttämättä $|S_\delta| \leq |T_{\beta N}|$.

Toisaalta, koska Lemman 4.2.6 nojalla $P(S) > 2^{-N(H(X)+\beta}$ kaikilla $S \in T_{\beta N}$ ja $P(T_{\beta N}) \leq 1$, niin $|T_{\beta N} \leq 2^{N(H(X)+\beta)}$ Täten

$$\begin{aligned} \frac{1}{N} H_\delta(X^N) &= \frac{1}{N} \log_2(|S_\delta|) \\ &\leq \frac{1}{N} \log_2(|T_{\beta N}|) \\ &\leq \frac{1}{N} \log_2(2^{N(H(X)+\beta)}) \\ &= \frac{1}{N}(N(H(X)+\beta)) \\ &= H(X) + \beta = H(X) + \epsilon. \end{aligned}$$

Täten väite siis pätee kaikilla tarpeeksi suurilla N.

Osa 2: $\frac{1}{N} H_\delta(X^N) \geq H(X) - \varepsilon$

Oletetaan, että väite ei päde, eli että on olemassa S_δ jolle $\frac{1}{N} \log_2(|S_\delta|) < H(X) - \varepsilon$. Valitaan $\beta = \varepsilon/2$, jolloin erityisesti $|S_\delta| < 2^{N(H(X)-2\beta)}$. Edelleen muistetaan, että Lemman 4.2.6 nojalla $P(\mathbf{x}) < 2^{-N(H-\beta)}$

kaikilla $\mathbf{x} \in T_{\beta N}$. Nyt

$$\begin{aligned} P(\mathbf{x} \in S_\delta) &= \overbrace{P(\mathbf{x} \in S_\delta \cap T_{\beta N})}^{\leq |S_\delta| \cdot 2^{-N(H(X)-\beta)}} + \overbrace{P(\mathbf{x} \in S_\delta \cap \complement T_{\beta N})}^{\leq \frac{\sigma^2}{\beta N}} \\ &\leq 2^{N(H(X)-2\beta)} \cdot 2^{-N(H-\beta)} + \frac{\sigma^2}{\beta N} \\ &= 2^{-N\beta} + \frac{\sigma^2}{\beta N}. \end{aligned}$$

Tästä seuraa, että $P(\mathbf{x} \in S_\delta) \to 0$ kun $N \to \infty$, mikä on ristiriita sillä oletuksen nojalla $P(\mathbf{x} \in S_\delta) \geq 1 - \delta$.

Nyt valitsemalla tarpeeksi iso N_0 ja $N \geq N_0$, pätevät kummatkin epäyhtälöt ja väite on siis todistettu. □

4.2.2 Kraf-McMillanin epäyhtälö

Lause 4.2.8 (Kraft-McMillan)**.** *Olkoon X TN-jakauma.*

1. *Jos c on yksikäsitteisesti purettavissa oleva binäärikoodi, niin tällöin*

$$\sum_{x \in \Omega_X} 2^{-\ell(c(x))} \leq 1.$$

2. *Jos on annettuna pituudet l_i, $i = 1, \ldots, |\Omega_X|$, siten että $\sum_i 2^{-l_i} \leq 1$, niin tällöin on olemassa yksikäsitteisesti purettavissa oleva etuliitevapaa koodi $\tilde{c}$ siten, että $\ell(c(a_i)) = l_i$ kaikilla $a_i \in \Omega_X$.*

Todistus. Todistetaan väitteet erikseen.

1. Merkitään $\{a_1, \ldots, a_n\} := \Omega$, $M := \sum_{j=1}^n 2^{-\ell_j}$ ja oletetaan että N on hyvin iso.

   Nyt

$$M^N = \left(\sum_{j=1}^n 2^{-\ell_j} \right) = \underbrace{\sum_{j=1}^n \sum_{j=1}^n \cdots \sum_{j=1}^n}_{N \text{ kpl}} 2^{-\ell_{i_1} - \ell_{i_2} - \ldots - \ell_{i_N}}$$

   Huomaa, että summaamme siis yli kakkosten potenssien, missä kakkosten potenssit ovat muotoa $-L$, missä L on jonkin N:n mittaisen merkkijonon koodauksen pituus.

Nyt kirjoitetaan summaus uuteen muotoon, missä summa on jaoteltu luvun L mukaan, eli

$$\underbrace{\sum_{j=1}^{n}\sum_{j=1}^{n}\cdots\sum_{j=1}^{n}}_{N \text{ kpl}} 2^{-\ell_{i_1}-\ell_{i_2}-\ldots-\ell_{i_N}} = \sum_{L=N\ell_{\min}}^{N\ell_{\max}} A_L 2^{-L},$$

Missä

$$\ell_{\min} = \min_{a_i\in\Omega} \ell_i, \quad \ell_{\max} = \max_{a_i\in\Omega} \ell_i,$$

ja

$$A_L = \#\{S \in \mathcal{M}_{\{0,1\}}^N : |c(S)| = L\}.$$

Nyt, koska oletimme että koodi on yksikäsitteisesti purettavissa, on on oltava $A_L \leq 2^L$, joten erityisesti saamme että

$$\sum_{L=N\ell_{\min}}^{N\ell_{\max}} A_L 2^{-L} \leq \sum_{L=N\ell_{\min}}^{N\ell_{\max}} 1 \leq N\ell_{\max} - N\ell\text{min} \leq N\ell_{\max}.$$

Täten kaikilla N pätee $M^N \leq \ell_{\max}N$. Nyt jos pätisi $M > 1$, niin termi M^N kasvaisi eksponentiaalisesti luvun N suhteen vaikka sillä on lineaarinen yläraja $\ell_{\max}N$. Tämä on ristiriita, joten oltava $M \leq 1$ kuten toivottiin.

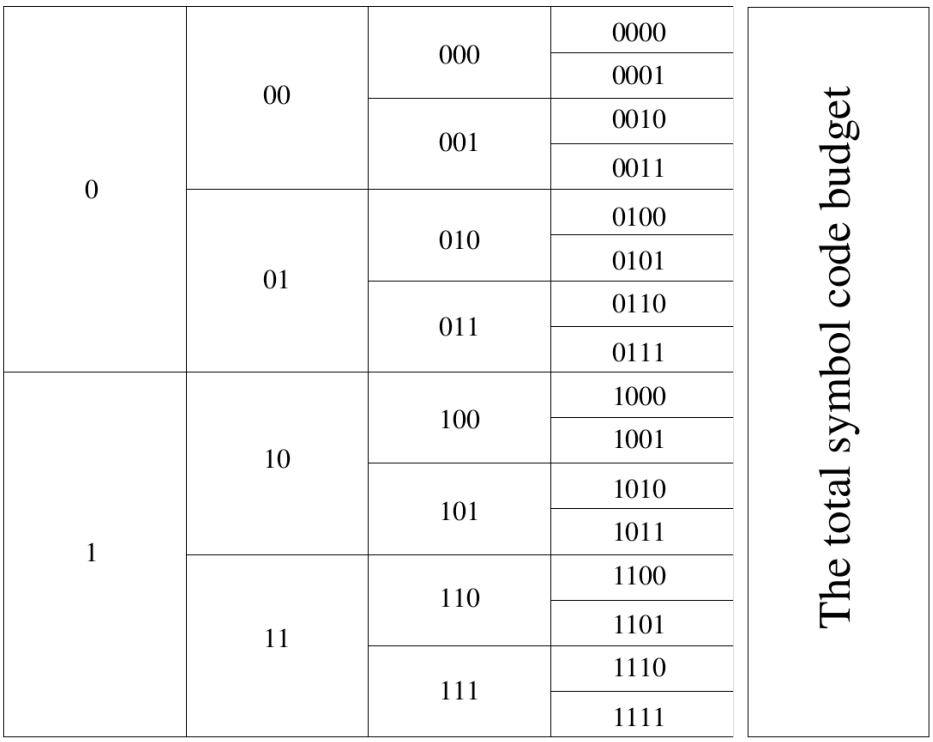
2. Tuijota kuvaa 4.2. Lause väittää, että mikäli sinulla on muotoa $\frac{1}{2^k}$ olevia lukuja $r_1, \ldots, r_n$ jotka summautuvat enintään ykköseksi, niin voit valita oheisten tyylisestä binääripuusta laatikoita $a_1, \ldots, a_n$ siten, että (i) jokaisen laatikon a_j korkeus on r_j (ii) minkään laatikon vasemmalla tai oikealla puolella ei ole toista valittua laatikkoa. Tämä onnistuu kun luvut r_j laitetaan laskevaan suuruusjärjestykseen, valitaan ensimmäinen laatikko siten, että se koskettaa koko budjetin yläreunaa, ja seuraava aina siten, että sen ja aiemman laatikon väliin ei jää yhtään tyhjää tilaa.

   Formalisointi jätetään harjoitustehtäväksi, katso myös [Mac03, Excercise 5.14, p.95].

□

Loppukaneettina, huomaa että aiemmin todistettiin että kaikilla koodeilla pätee $L(c, X) \geq H(X)$. Se, mitä kyseessä oleva todistus kuitenkin tarkalleen ottaen antaa on

$$L(c, X) \geq H(X) + D_{KL}(P||Q) - \log_2(z_0),$$



Kuva 4.2: "Bittibudjetti"Kraft-MacMillanin epäyhtälön todistuksessa.
Lähteestä MacKay [Mac03].

missä P on jakauman alkuperäinen todennäköisyysjakauma, $z_0 = \sum_j 2^{-\ell_j}$ ja Q on jakauma joka määritellään asettamalla $q_i = 2^{-\ell_i}/z_0$. (Huomaa, että Kraft-McMillanin nojalla $z_0 \leq 1$.

Koodin odotuspituus on siis sitä pienempi mitä lähempänä z_0 on ykköstä (eli mitä 'tiiviimmin' viestipituuden on valittu) ja mitä 'lähempänä' alkuperäinen jakauma P on jakaumaa Q Kullback-Leibler -divergenssin mielessä.

Luku 5

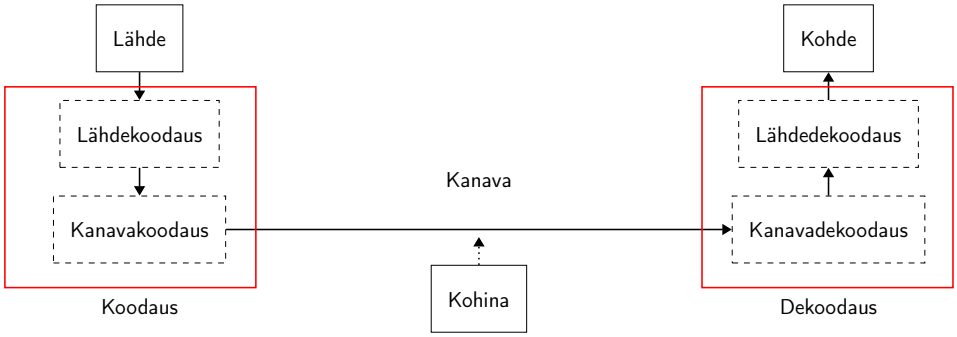
Koodausteoriaa kohinalla

Edellisessä kappaleessa tutkittiin kohihattomalla kanavalla kommunikointia. Tässä kappaleessa lisätään tilanteeseen mahdollisen kohinan vaikutus. Kohinattomassa tilanteessa pyrimme tiivistämään lähetettävät viestit mahdollisimman lyhyiksi nopeuttaaksemme viestintää, eli teimme kompressiota, eli poistimme redundanssia. Kohinaisessa tilanteessa tavoite on edelleen nopea viestintä, mutta haluamme myös varautua mahdollisiin lähetyksen aikana syntyviin virheisiin, ja käytännössä tämä tarkoittaa redundanssia lisäämistä.

Kohinaisessa kommunikaatiossa tilanne siis näyttää siltä, että yhtä aikaa poistamme ja lisäämme redundanssia. Ideana onkin, että ensin poistamme viestistä 'suunnittelemattoman' redundanssin ja tämän jälkeen lisäämme siihen tarkoitushakuista, virheensietokykyä optimoivaa redundanssia.

5.1 Perusosa

Yleisasetelma on sama kuin viime luvussa. Tilanteen yksinkertaistamiseksi rajoitumme tilanteeseen missä kanava on *binäärikanava*, eli siellä kulkee ykkösiä ja nollia. Edelleen jaamme suorittamamme koodauksen kahteen osaan: kompressointiin eli *lähdekoodaukseen* sekä virheensietokyvyn lisäämiseen eli *kanavakoodaukseen*. Viime osion havainnakuva muokkautuu siis seuraavan näköiseksi.



Havainnekuva kohinaisesta kanavasta.

Oletamme edelleen, että lähdekoodaus on tehty optimaalisesti, eli että lähteestä tulevaa dataa ei voi enää edelleen kompressoida, ja että lähdekoodauksen output on binäärisymboleita. Erityisesti lähdekoodaus on meille black box, ja mallinnamme sitä TN-jakaumalla $((0,1),(p_0,p_1))$. Huomataan heti, että jos tämä jakauma ei ole tasajakauma, on sen entropia alle 1, jolloin viime luvun tekniikoilla voisimme edelleen kompressoida viestejä paremmin. Täten voimme siis optimaalisuusoletuksen nojalla olettaa, että $p_0 = 0.5 = p_1$.

Kertauksen vuoksi, oletamme tässä kappaleessa siis että:

- Kanavamme on binäärikanava.
- Koodaus on jaettu lähde- ja kanavakoodauksiin.
- Lähdekoodaus on optimaalinen binäärikoodi, ja mallinnamme sitä tasajakaumalla.

Emme puutu tällä kurssilla siihen, että miten rajoittavia nämä oletukset ovat. Huomautamme tosin, että todistukset näyttävät hyvin samoilta vaikka kanava olisi ei-binäärinen.

5.1.1 Kohinainen kanava

Määritellään seuraavaksi kanava. Kanava on siis jokin kommunikaatioväylä, kuten puhelinlinja, parikaapeli, radioyhteys tai vastaava, johon voi jollain todennäköisyydellä tulla virheitä. Määrittelemme kanavan parina TN-

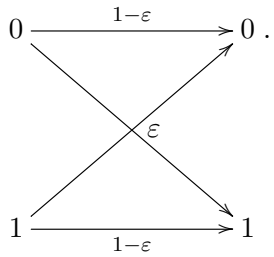
jakaumia, jotka kuvastavat miten todennäköisesti vastaanottopäässä nähdään 0 tai 1 kun on lähetetty 0 tai 1.

Määritelmä 5.1.1. *Kanava* on pari TN-jakaumia (Y_0, Y_1), missä

$$Y_0 = ((0,1),(p_0,p_1)) \quad \text{ja} \quad Y_1 = ((0,1),(q_0,q_1)).$$

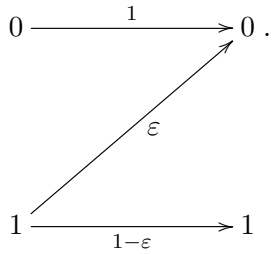
Esimerkki 26. Tärkein esimerkki meille on *symmetrinen ε-kohinainen kanava* $(Y_0^\varepsilon, Y_1^\varepsilon)$, missä $p_0 = 1-\varepsilon$, $p_1 = \varepsilon$, $q_0 = \varepsilon$ ja $q_0 = 1-\varepsilon$.

Sen sijaan että kirjoittaisimme kaikki parin todennäköisyydet aina näkyviin, esitämme kanavia usein seuraavassa muodossa:



Tässä kanavassa siis kummallakin syötteellä on sama todennäköisyys välittyä oikein vastaanottajalle.

Esimerkki 27. Toinen tärkeä esimerkki on n.k. ε-kohinainen Z-kanava:



Tässä kanavassa viesti 0 välittyy siis aina oikein, mutta viesti 1 voi välittyä väärin.

Tulemme 'syöttämään' kanavalle kanavakoodauksestamme syntyviä bittijonoja. Kanavan 'näkökulmasta' kanavakoodauksemme output on TN-jakauma. Kanava ja sille syötteenä toimiva TN-jakauma muodostavat yhteisjakauman seuraavasti.

Määritelmä 5.1.2. Olkoon (Y_0, Y_1) kanava ja $X := ((0,1),(t_0,t_1))$ TN-jakauma. Näiden jakaumista muodostetaan yhteisjakauma (X,Y) asettamalla

$$P(X=0,Y=0) = t_0p_0, \quad P(X=0,Y=1) = t_0p_1,$$
$$P(X=1,Y=0) = t_1q_0, \quad P(X=1,Y=1) = t_1q_1.$$

Tätä yhteisjakaumaa kutsutaan *kanavajakaumaksi*.

Huomaa, että kanavajakaumassa $P(X|Y = 0) = Y_0$, $P(X|Y = 1) = Y_1$. Merkitsemme välillä röyhkeästi sekä kanavaa (Y_0, Y_1) että sen kanssa muodostetun kanavajakauman marginaalitodennäköisyyttä samalla merkillä Y.

Kohinaisella kanavalla kiinnostavaa on se, miten hyvin voimme päätellä lähetetyn bitin kun tiedämme vastaanotetun bitin. Erityisesti meitä siis kiinnostaa miten vahvasti kanavajakauman marginaalijakaumat ovat kytköksissä toisiinsa. Tätä varten meillä on jo olemassa loistava mittari, nimittäin yhteisinformaatio. Muistetaan, että yhteisinformaatio voidaan laskea kaavalla

$$I(X;Y) = H(X) - H(X|Y).$$

Tässä kohtaa kurssia on hyvä pitää mielessä tulkinta, jossa $H(X)$ kuvaa ylipäänsä epätietoisuuttamme lähetetystä bitistä, ja ehdollinen entropia $H(X|Y)$ sitä, että kuinka epätietoisia olemme keskimäärin bitistä lähetetystä bitistä kun olemme nähneet saapuneen bitin. Voitaisiin siis kirjoittaa

Yhteisinformaatio = Keskim. epävarmuus alussa − Keskim. epävarmuus lopussa.

Yhteisinformaatio siis luontevasti kuvaa paljon voimme yleensä toivoa saavamme tietoa viestin vastaanotettuamme. Tämän motivoimana voidaan määritellä kanavan kapasiteetti.

Määritelmä 5.1.3. Kanavan Y *kapasiteetti* on

$$\mathrm{Cap}(Y) = \max_X I(X;Y),$$

missä maksimi otetaan kaikkien kanavan Y kanssa muodostettujen kanavajakaumien yli.

Shannonin kanavakoodauslause, josta myöhemmin puhumme, kertoo että kanavan kapasiteetti on nimensä veroinen, eli kapasiteetti antaa jossain mielessä tarkan rajan sille, miten nopeasti kanavalla voi edes teoreettisesti viestiä.

Demoissa lasketaan symmetrisen ε-kohinaisen binäärikanavan kapasiteetti.

5.1.2 Kanavakoodaukset

Viime kappaleessa määrittelimme TN-jakaumaan (Ω_X, P) liittyvän koodauksen funktioksi $c\colon \Omega_X \to \mathcal{M}_{0,1}$. Tämän jälkeen suunnittelimme blokkikoodeja muokkaamalla alkuperäisestä jakaumasta X jakauman X^N, ja määrittelemällä tälle jonkin koodauksen. Olisimme voineet vaihtoehtoisesti määritellä, että koodauksen määrittelyjoukko saa olla isompi kuin $\Omega_X = \mathcal{M}^1_{\Omega_X}$, erityisesti koodauksen voisi määritellä funktioksi $c\colon \mathcal{M}^{N*}_{\{0,1\}} \to \mathcal{M}_{\{0,1\}}$, jollain

$N \geq 1$, missä $\mathcal{M}_{\{0,1\}}^{N*}$ on kaikkien tasan N merkkiä pitkien binäärisanojen joukko. Teemme näin jatkossa.

Erityisesti jos koodaus on muotoa $c\colon \mathcal{M}_{\{0,1\}}^{K*} \to \mathcal{M}_{\{0,1\}}^{N}$ jollain $N, K \geq 1$, niin kutsumme tällaista koodausta (N,K)-*(blokki)koodaukseksi.* (Keskitymme pelkästään blokkikoodeihin, eli erityiesti emme mieti vaihtelevan pituisia koodeja kanavakoodauksessa. Yksi syy on se, että etuliitevapaan koodit ovat vielä ihan eri tavalla alttiita häiriöille, ja koodisanojen rajojen häilyminen voisi aiheuttaa isoja virheitä.) Blokkikoodauksen *lähetysnopeus* on luku K/N, eli esimerkiksi jos kanava pystyy välittämään yhden merkin sekunnissa, niin käytettäessä blokkikoodia jonka nopeus on R voidaan viestiä keskimäärin R kanavakoodauksen lähteen merkkiä sekunnissa.

Koodaukseen liittyy aina myös koodauksen purku. Kohinattomassa tilanteessa koodin purku oli suoraviivaista, mutta kohinaisessa tilanteessa meidän pitää varautua virheisiin. Mikäli c on (N,K)-koodaus niin sen purku (eli dekoodaus) on funktio $d\colon \mathcal{M}_{\{0,1\}}^{N*} \to \mathcal{M}_{\{0,1\}}^{K}$ jolle pätee $d(c(x)) = x$. Huomaa, että käytännössä aina $N > K$, joten koodin purku antaa saman tuloksen useammalle eri syötteelle. Vertaa halutessasi tilannetta lineaarikuvauksiin $\mathbb{R}^K \to \mathbb{R}^N$ yms.

Seuraava määritelmä on hieman tekninen, mutta sen taustalla on yksinkertaisia ajatuksia. Erityisesti, **symmetrisen kohinaisen kanavan tilanteessa esittelemämme eri virhekäsitteet ovat kaikki samoja, ja erityisesti "todennäköisyys virheelle viestissä" on juuri se, mitä sen arvaisitkin olevan.** Jos se ahdistaa, katso joko myöhempiä esimerkkejä jossa virheitä lasketaan, tai perehdy MacKayn kirjan lukuun 9.

Määritelmä 5.1.4. Olkoon Y kanava ja c (N,K)-koodaus sekä d sen jokin purku. Jos T on kanavalle lähtevä merkkijono, niin merkitsemme $\Xi(T)$ merkkijonoa, joka otetaan kanavan toisessa päässä vastaan – huomaa että c ja d ovat deterministisiä funktioita, mutta kanavalla on stokastinen luonne, joten Ξ on nyt välttämättä satunnaismuuttuja. Mutta minkä suhteen?

Koodauksemme c on itseasiassa satunnaismuuttuja joka on määritelty TN-jakaumassa $((0,1),(0.5,0.5))^K$ (muista, että oletimme että kanavakoodauksen input on luonteeltaan kahden alkion tasajakauma optimaalisen kompression johdosta). Täten kanavakoodauksen output muodostaa uuden TN-jakauman kun sen ajatellaan tuottavan bittejä yksi kerrallaan. Tämä TN-jakauma puolestaan muodostaa kanavan Y kanssa kanavajakauman, jonka avulla voimme viimein laskea todennäköisyyden sille, että annettu viesti välittyy kanavalla oikein. Emme kirjoita näitä kuitenkaan formaalisti auki, ainakaan perusosassa, vaan merkitsemme merkillä Ξ satunnaismuuttujaa joka kuvaa sitä mitä kanava tekee syötteilleen (eli jos kanavaan työntää merkkijonon T niin ulos tulee $\Xi(T)$ ja merkitsemme $P(d(\Xi(c(S))) \neq S)$ todennäköisyyttä sille, että annetuilla (deterministisillä) koodauksella ja purulla saamme takaisin väärän viestin kun olemme koodanneet, lähettäneet kanavalle ja dekoodanneet.

Tästä edelleen määritellään:

- *Keskimääräinen blokkivirhe* p_B on todennäköisyys sille että lähetetty merkkijono purkautuu väärin.

$$p_B = \sum_{S \in \mathcal{M}_{0,1}^K} P(S) \cdot P(d(\Xi(c(S))) \neq S)$$

- *Suurin mahdollinen blokkivirhe* p_{BM}

$$\max_{S \in \mathcal{M}_{0,1}^K} P(d(\Xi(c(S))) \neq S)$$

- *Bittivirhe* p_b on keskimääräinen todennäköisyys sille, että alkuperäisen viestin yksittäinen bitti välittyy määränpäähän väärin.

- Purku on *optimaalinen* mikäli se minimoi keskimääräisen blokkivirheen koodauksen kaikkien purkujen joukossa. (Huomaa, että tämä voi riippua kanavasta.)

Symmetrisellä kohinaisella kanavalla virheen todennäköisyys ei riipu syötteestä, joten kaikissa tutkimissamme esimerkeissä kaikilla syötteillä on sama virhetodennäköisyys ja oletustemme nojalla kaikilla syötteillä sama todennäköisyys, jolloin $p_B = p_{BM} = p_b$.

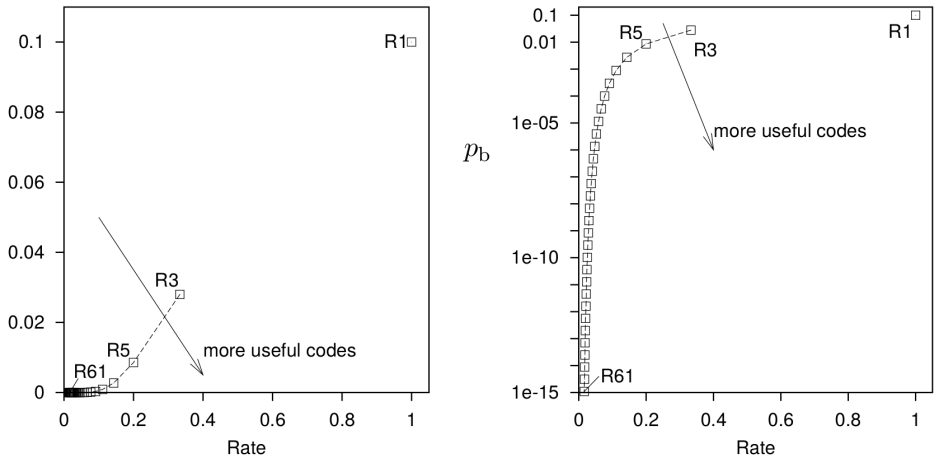
Otetaan nyt muutama esimerkki koodauksista.

Esimerkki 28. Koodi $R_3 \colon \mathcal{M}_{\{0,1\}}^{1*} \to \mathcal{M}_{\{0,1\}}^{3*}$ määritellään asettamalla $0 \mapsto 000, 1 \mapsto 111$. Tämä koodi siis triplaa kaikki merkit. Koodi puretaan vastaanottopäässä katsomalla kutakin kolmen merkin blokkia, ja katsomalla kumpaa merkkiä on enemmän, eli koodin purku vie sanat $111, 101, 110$ ja 011 merkille 1 ja loput sanat merkille 0. Tämä on $(3, 1)$-koodaus.

Oletetaan, että lähetämme tällä koodauksella viestin 1010 symmetrisellä $\varepsilon = 0.1$-kohinaisella kanavalla. Mikä on todennäköisyys, että viesti tulee virheettä perille? Esimerkin tilanteessa kukin merkki lähetetään kolminkertaisena, ja purku onnistuu mikäli enintään yksi merkki vaihtuu, joten todennäköisyys että saamme virheen on

$$P(\text{kaksi virhettä}) + P(\text{kolme virhettä}) = 3 0.1^2 0.9^1 + 0.1^3 \approx 0.03.$$

Verrattuna identtiseen koodaukseen missä kukin symboli vaan lähetetään itselleen (eli $(1, 1)$-koodaus) ja virheen todennäköisyys on $\varepsilon = 0.1$, saamme nyt pienennettyä virheen todennäköisyyden alle kolmannekseen alkuperäisestä. Huonona puolena on se, että joudumme lähettämään kolme kertaa enemmän merkkejä kuin ennen, eli *lähetysnopeutemme tippuu kolmannekseen alkuperäisestä.*



Kuva 5.1: Muutamien koodien virheensietokyvystä ja lähetysnopeudesta. Kummassakin x-akselina on lähetysnopeus ja y-akselina blokin keskimääräinen virhe, vasemmassa lineaarisella asteikolla ja oikeassa logaritmisella. Kuva lähteestä [Mac03, s. 8].

Vastaavasti voitaisiin määritellä $(5,1)$-koodi R_5, jossa $0 \mapsto 00000, 1 \mapsto 11111$ jolloin blokin keskimääräinen virhe on todennäköisyys sille että sanomaan osuu vähintään kolme virhettä;

$$\binom{5}{3}0.1^3 0.9^2 + \binom{5}{4}0.1^3 0.9^2 + 0.1^5 \approx 0.0087.$$

Virheen todennäköisyys siis laskee entisestään, mutta myös lähetysnopeus pienenee huomattavasti.

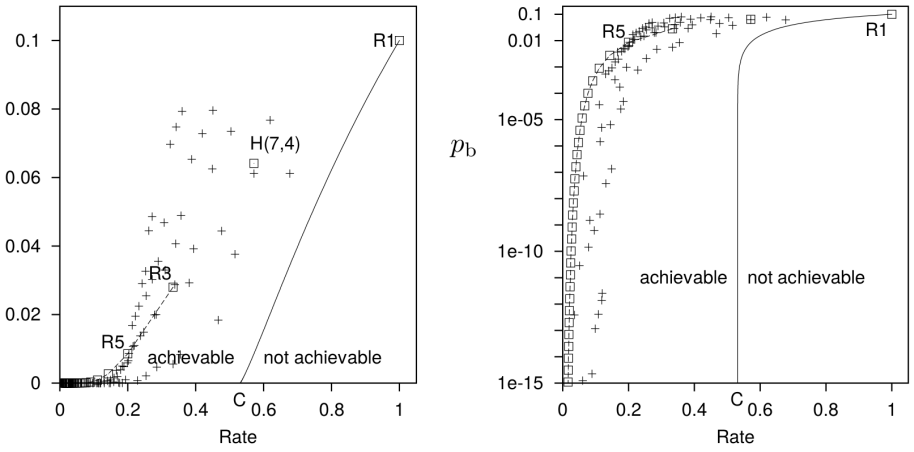
Edelleen voitaisiin määritellä $(k,1)$-koodit R_k, missä k on mikä tahansa pariton luonnollinen luku. Oheisessa MacKayn kirjasta napatussa kuvassa (Kuva 5.1) on plotattu näiden koodien antamaa yksittäisen lähetetyn merkin virhettä suhteessa niiden lähetysnopeuteen.

Yllä käytetyt R_k-koodit ovat luontevan oloisia, joten tässä kohtaa voisi arvata, että ne kuvaavat yleistä tilannetta: parempi virheensietokyky vaatii aina vaan suurempaa lähetysnopeuden uhraamista. Näin ei kuitenkaan ole! Shannonin kanavakoodauslause sanoo seuraavaa.

Lause 5.1.5. *Olkoon Y kanava ja C sen kapasiteetti. Tällöin Kaikilla $\varepsilon > 0$, $R < C$ ja tarpeeksi isoilla*[1] *N on olemassa (N,K)-blokkikoodi jonka lähetysnopeus on vähintään R ja blokin suurin mahdollinen virhe alle ε.*

Todistus. Torstaina. □

[1] 'Tarpeeksi iso' riippu kanavasta sekä parametreista ε ja R.



Kuva 5.2: Muutamien koodien virheensietokyvystä ja lähetysnopeudesta. Kummassakin x-akselina on lähetysnopeus ja y-akselina blokin keskimääräinen virhe, vasemmassa lineaarisella asteikolla ja oikeassa logaritmisella. Kuvassa näkyy myös Shannonin kanavakoodauslauseen antama raja. Kuva lähteestä [Mac03, s. 15].

Tämä lause siis sanoo, että meidän ei tarvitse uhrata viestintänopeutta alle tietyn kanavasta riippuvan vakion saadaksemme virhetodennäköisyydet todella pieniksi. Pienenä uhrauksena joudumme tosin käyttämään aina vaan pidempiä ja pidempiä viestiblokkeja, joka voi tuottaa ongelmia mikäli meillä olisi tarve viestiä silloin tällöin lyhyitä viestejä eikä jatkuvaa datavirtaa. Tähän liittyvän kommentaarin jätämme koodausteorian ammattilaisille. Shannonin lausetta on havainnoitu kuvassa 5.2.

Shannonin lauseen todistus jätetään syventävään osioon, mutta tutustutaan seuraavaksi Hammingin koodiin jota tutkimalla huomaamme, että miksi R_k-koodit eivät pääse lähellekään optimaalista tilannetta.

5.1.3 Hammingin koodi

Miksi R_k-koodit eivät ole optimaalisia? Tarkastellaan koodia tutkimalla merkkijonon $S = 1010$ lähetystä. R_3 triplaa kunkin yksittäisen merkin, eli kanavalle lähtee merkkijono $T = 111000111000$. Huomataan, että koodi osaa korjata minkä tahansa yksittäisen virheen, eli mikäli $T*$ on merkkijono joka saadaan jonosta T yhtä merkkiä muuttamalla, niin $d_3(T*) = S$, missä d on koodauksen R_3 purku.

Huomataan, että todennäköisyys saada tasan yksi virhe viestin T lähetyksessä on $12 * 0.1 * 0.9^{11} \approx 0.38$. Koodaus osaa kuitenkin korjata myös muita virheitä, esimerkiksi virheen jossa lähetyksessä syntyy neljä virhettä,

mutta kukin sijoittuu eri 'triplablokkiin' lähetetyssä viestissä:

$$S = \texttt{1010} \xrightarrow{R_3} T = \texttt{111000111000}$$
$$\texttt{--}\star\star\texttt{---}\star\texttt{---}\star$$
$$S = \texttt{1010} \xleftarrow{d} T' = \texttt{110100101001}$$

Todennäköisyys saada lähetyksen aikana tällainen virhe on $0.1^4 \cdot 0.9^8 \cdot 1 \cdot \frac{3}{4} \cdot \frac{2}{4} \cdot \frac{1}{4} \approx 0.000004$. Tämä on huiman pieni, kun vertaa sitä todennäköisyydeen saada kaksi virhettä samassa blokissa, eli $0.1^2 \cdot 0.9^{10} \cdot 1 \cdot \frac{2}{11} \approx 0.0006$.

Toisin sanoen, koodaus R_3 osaa korjata erilaisia virheitä, mutta ongelma on siinä että sitä ei ole rakennettu korjaamaan parhaiten todennäköisimpiä virheitä. Tilanne on hieman samanlainen kuin Huffmanin koodista keskustellessa; lähdekoodauksessa kannattaa antaa lyhyitä koodeja kaikista todennäköisimmille merkeillä. Vastaavasti kanavakoodauksessa kannattaa luoda koodeja jotka korjaavat parhaiten todennäköisimpiä virheitä.

Lähdekoodaus:	**Kanavakoodaus:**
Kaikkea ei voi kompressoida.	Virhettä ei saa ikinä nollaksi.
Keskimääräinen parannus riittää.	Keskimääräinen parannus riittää.
Annetaan siis yleisemmille asioille parempi kompressio.	*Annetaan siis yleisemmille virheille parempi korjaus.*

Hammingin koodin ideana onkin juuri tämä; se osaa korjata tosi hyvin yksittäisiä virheitä eikä käytä "energiaa" harvinaisempien virheiden korjaamiseen.

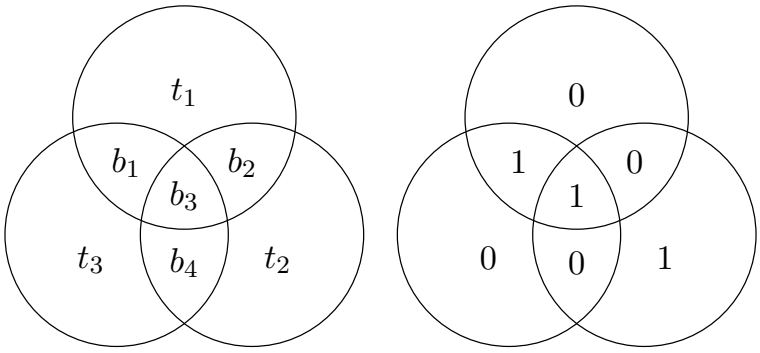
Hammingin koodaus

Hammingin $(7, 4)$-koodi on $(7, 4)$-koodaus, joka määritellään seuraavasti. Olkoon $S = b_1b_2b_3b_4 \in \mathcal{M}_{0,1}^{4*}$ merkkijono, joka halutaan kooodata. Haluamme määritellä seuraavaksi nk. *parillisuusbitit* $t_1t_2t_3$; lopullinen koodi tulee olemaan $b_1b_2b_3b_4t_1t_2t_3$. Piirretään kolme ympyrää kuten kuvassa 5.3, ja **valitaan parillisuusbitit siten, että kussakin ympyrässä on parillinen määrä ykkösiä.**

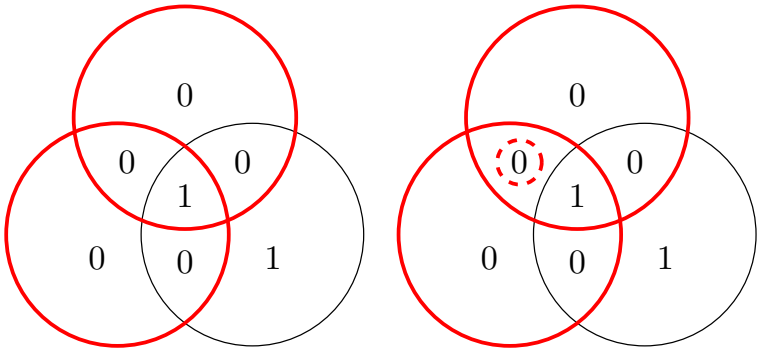
Esimerkillä 1010 lähetetään siis viesti 1010010.

Hammingin purku

Hammingin koodia purettaessa, jos matkalla ei ole tapahtunut yhtään virhettä, löytyy alkuperäinen viesti ensimmäisestä neljästä merkistä. Viestiä vastaanotettaessa pitää kuitenkin tarkistaa, onko virheitä mahdollisesti tullut. Hammingin koodissa tämä tehdään katsomalla, että onko kaikissa ympyröissä edelleen parillinen määrä ykkösiä. Jos on, niin asetetaan $d(b_1b_2b_3b_4t_1t_2t_3) = b_1b_2b_3b_4$.



Kuva 5.3: Hammingin koodin määrittely



Kuva 5.4: Hammingin koodin purkua virheellisellä viestillä $T* = 0010010$.

Mikäli ei ole, niin katsotaan missä ympyröissä on pariton määrä ykkösiä. Tällöin on olemassa yksikäsitteinen bitti jota muuttamalla kaikkien ympyröiden ykkösten määrä palaa parilliseksi. Katsotaan tilannetta kahden esimerkin kautta. Tutkitaan tilanteita, joissa aiemmin koodattuun viestiin 1010010 on tullut virhe joko ensimmäiseen bittiin $S* = 0010010$ tai viimeiseen bittiin $T* = 1010011$. Purku näkyy kuvissa 5.4 ja 5.5.

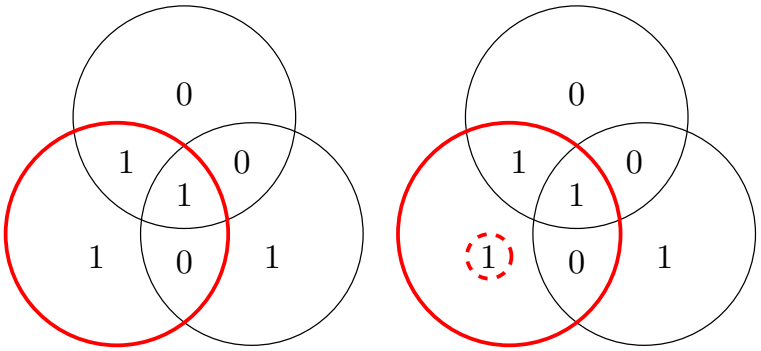
Demoissa tutkitaan mitä käy useamman virheen tilanteessa.

Hammingin koodi matriiseilla

Mikäli varustetaan joukko $\{0, 1\}$ laskutoimituksella

$$0+0=0 \quad 0+1=1$$
$$1+0=1 \quad 1+1=0$$

(eli XOR tai mod 2 -aritmetiikka, as you wish) niin huomataan että bittien $\{a, b, c, d\}$ joukossa on parillinen määrä ykkösiä jos ja vain jos $a+b+c+d = 0$. Erityisesti jos merkataan $t = a+b+c+d$, niin joukossa $\{a, b, c, d, t\}$ on aina parillinen määrä ykkösiä.



Kuva 5.5: Hammingin koodin purkua virheellisellä viestillä $T* = 1010011$.

Täten Hammingin koodi voidaan määritellä asettamalla $t_1 = b_1 + b_2 + b_3$ jne. Käyttämällä lineaarialgebrasta tuttua notaatiota, voidaan Hammingin koodaus siis ilmaista matriisin muodossa:

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} \mapsto \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_1 + b_2 + b_3 \\ b_2 + b_3 + b_4 \\ b_1 + b_3 + b_4 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ t_1 \\ t_2 \\ t_3 \end{bmatrix},$$

jossa yhteenlaskut siis tulkitaan aiemmin mainitussa muodossa.

5.2 Syventävä laajennos

Tässä kappaleessa käydään läpi Shannonin kanavakoodauslauseen todistuksen ideaa. Emme mene kovin syvälle yksityiskohtiin, joita on paljohkosti, vaan kuvailemme todistuksen suuret linjat. Seuraamme pitkälti MacKayn kirjan [Mac03] luvun 10 esitystä, josta lukija voi halutessaan käydä tarkemmat yksityiskohdat läpi. (Jälleen, katso myös [Sha48].)

Karkeana ideana lauseessa on taas hyödyntää pitkiä binääriblokkeja. Tulemme siis muodostamaan muodostamaan binäärisymboleita $\{0, 1\}$ lähettävästä kanavasta Y binääriblokkeja $\mathcal{M}_{0,1}^{N*}$ lähettävän kanavan Y^N. Tämän jälkeen, hieman kuten lähdekoodauslauseen kanssa, rajoitumme käyttämään vaan pientä osaa mahdollisista blokeista $\mathcal{M}_{0,1}^{N*}$. Tämä osajoukko, olkoon K, on sellainen, jota voi viestiä lähes virheettä.

Tällä systeemillä voimme siis viestiä, lähettämällä N kokoisia blokkeja, K erilaista viestiä, eli voimme lähettää sillä $\lfloor \log_2(K) \rfloor$ pituisia binäärijonoja. Käykin ilmi, että lukumäärältään suurin K mitä voimme valita on noin $2^{N \operatorname{Cap}(Y)}$, eli juuri se, mitä Shannonin kanavakoodauslause ilmoittaa mahdolliseksi.

Viestintitä'protokolla' näyttää siis suunnilleen tältä:

1. Sinulla on annettuna kanava Y ja viestilähde Ω (binäärinen tasajakauma).
2. Päätät miten pienen virheen ε haluat sallia.
3. Nokkelilla menetelmillä etsit blokkikoon N siten, että virhe voidaan puskea teoriassa alle ε.
4. Muodostat N-kanavan Y^N ja valitset, jälleen nokkelasti, $M := 2^{N \operatorname{Cap}(Y)}$ viestiä, joita tulet kanavalla lähettämään.
5. Muodostat lähdejakaumasta $K = \log_2(M)$-kertaisen jakauman, jonka alkioille valitset kullekin jonkin annetuista viestiblokeista.
6. Teippaat kaiken yhteen ja vot!

Pienenä Caveattina, Shannonin kanavakoodauslause kertoo ainoastaan mikä on teoreettisesti mahdollista ja mahdotonta. Tämä antaa myös käytännön tilanteisiin rajoitteita, mutta Shannonin lauseen todistus ei ole konstruktiivinen, eli se ei kerro miten onnistut luvun N ja viestiblokit valitsemaan. Käytännössä siis tämä protokolla ei ole välttämättä kovin käyttökelpoinen.

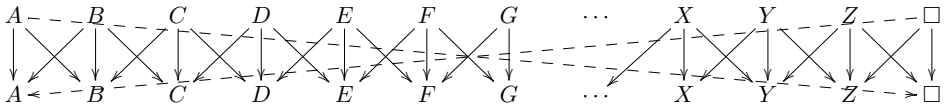
5.2.1 Kohinainen kirjoituskone

Tarkastellaan ensin Shannonin kanavakoodauslauseen sisältöä erityisen kanavan tapauksessa. Seuraava, nk. *kohinainen kirjoituskone* (noisy typewriter) on klassinen esimerkki Shannonin lauseen kuvailussa.

Kohinainen kirjoituskone on kanava $KK := (Y_A, Y_B, \ldots, Y_Z, Y_\square)$, missä kukin jakauma Y_α, $\alpha \in \{A, B, \ldots, Z, \square\}$ on kolmen alkion tasajakauma siten, että

$$Y_\alpha(\alpha) = Y_\alpha(\alpha - 1) = Y_\alpha(\alpha + 1) = \frac{1}{3}$$

kun tulkitaan että $B + 1 = C$, $Q + 1 = R$, $Z + 1 = \square$, $\square + 1 = A$ jne.



Tämä kanava kuvaa siis kirjoituskonetta, jossa painettu kirjain tuottaa joko oikean kirjaimen tai sitten aakkosissa sen vieressä olevan kirjaimen, kaikki samalla todennäköisyydellä. Lasketaan ensin kanavan kapasiteetti. Koska kanava on todella symmetrinen, huomaamme että kanavan maksimoivan jakauman pitää olla myös hyvin symmetrinen, erityisesti (sivuutamme detaljit, vapaaehtoinen ylimääräinen ht.)

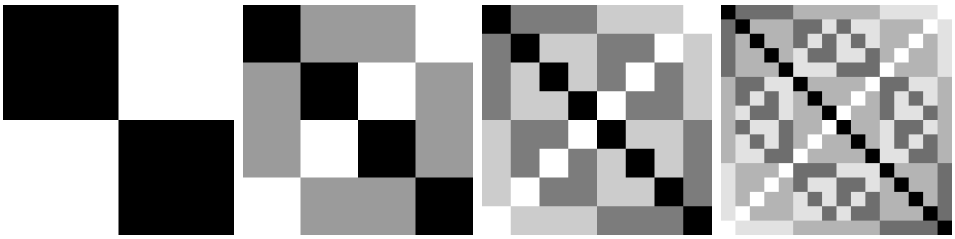
$$\mathrm{Cap}(Y) = I(P_U; Y),$$

eli löydämme kanavan KK kapasiteetin muodostamalla kanavajakauman aakkosten tasajakauman kanssa ja laskemalla tämän kanavajakauman marginaalijakaumien yhteisinformaation.

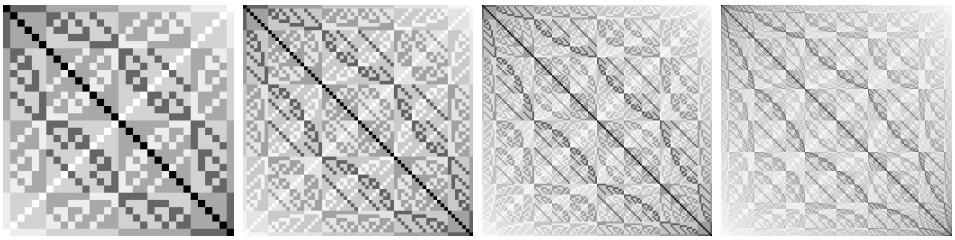
Toisaalta huomaamme, että kanavajakaumassa (X, Y) kullakin aakkosella α ehdollinen jakauma $(X|Y = \alpha)$ on aina kolmen alkion tasajakauma. Toisaalta nopealla laskulla huomataan että marginaalijakauma Y on myöskin tasajakauma. Täten

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= \log_2(27) - \sum_\alpha P(Y = \alpha) H(X|Y = \alpha) \\ &= \log_2(27) - \sum_\alpha \frac{1}{27} \log_2(3) \\ &= \log_2(9) \\ &\approx 3.2. \end{aligned}$$

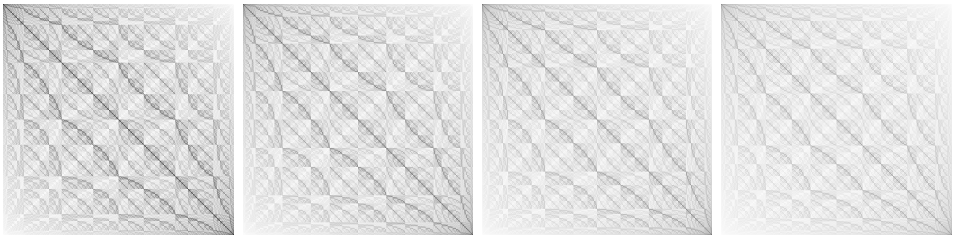
Nyt huomataankin, että Shannonin kanavalauseen nojalla tällä kanavalla pitäisi pystyä viestimään virheettä ainakin kolmen bitin nopeudella. Kolmen bitin nopeus onnistuu tässä erikoistapauksessa itseasiassa täysin virheettä: valitaan aakkosista joka kolmas, eli muodostetaan joukko $K = \{A, D, G, J, \ldots, V, Y\}$. Jos kommunikoimme kanavalla vain näitä symboleita, eivät mitkään kaksi näistä voi kuvautua samalle symbolille eli viestintä



(a) $N = 1$ (b) $N = 2$ (c) $N = 3$ (d) $N = 4$



(e) $N = 5$ (f) $N = 6$ (g) $N = 7$ (h) $N = 8$



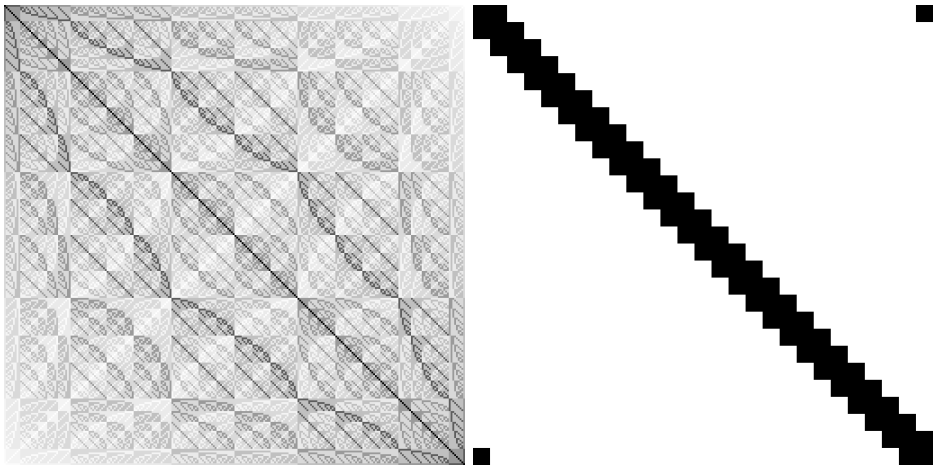
(i) $N = 9$ (j) $N = 10$ (k) $N = 11$ (l) $N = 12$

Kuva 5.6: Symmetrisen 0.1-kohinaisen kanavan Y tasajakauman X kanssa muodostettu N-kertainen yhteisjakauma, eli N-kertainen kanavajakauma. (Värit skaalattu s.e. pienin tn on valkoinen, suurin musta ja muut riippuvat monotonisesti todennäköisyydestä.)

on täysin virheetöntä. Näillä 9 kpl merkeillä voidaan lähettää $\lfloor \log_2(9) \rfloor = 3$ mittaisia merkkijonoja, eli tällä kanavalla viestinopeus on $9/3 = 3$, eli hyvin lähellä kapasiteetin antamaa rajaa.

Shannonin kanavakoodauslauseen idea on, että jossain mielessä kaikista kanavista saadaan kohinaisen kirjoituskoneen näköinen, kun kanavalla aletaan lähettää tarpeeksi pitkiä viestiblokkeja. Oheisessa kuvassa 5.6 näytetään, miltä symmetrinen ε-kohinainen kanava $\varepsilon = 0.1$ rupeaa näyttämään, kun sillä viestitään N mittaisia merkkijonoja. Kussakin kuvassa kukin pikseli (x, y) kuvaa todennäköisyyttä sille, että jos kanavalle lähetetään merkkijono x symboli kerrallaan, niin ulos tulee merkkijono y. (Bittijonot on järjestetty kasvavassa järjestyksessä ykkösten lukumäärän suhteen.) Värit on skaalattu s.e. suurin tn on musta, pienin valkoinen ja muut harmaasävyjä näiden välissä.

Kuva ei näytä päälle päin hirvesti konekirjoituskanavan vastaavalta ku-



(a) $N = 8$ (b) KK-jakauma

Kuva 5.7: Symmetrisen 0.1-kohinaisen kanavan Y tasajakauman X kanssa muodostettu N-kertainen yhteisjakauma, $N = 8$ ja kohinaisen konekirjoituskoneen jakauma.

valta, jossa 27 kertaa 27 pikselin kokoisessa kuvassa olisi kolmen pikselin levyinen musta raita keskellä ja yksittäiset mustat pikselit nurkissa, (kts. kuva 5.7) mutta siinä on samanlaisia tärkeitä ominaisuuksia. Erityisesti, todennäköisyys on 'pakkautunut' kapeille raidoille, joita voi käyttää kommunikoinnissa.

5.2.2 Viestiblokit ja tyypilliset merkkijonot

Lähdekoddauslauseen todistuksen yhteydessä, määritimme jakaumalle $X = ((0,1),(p_0,p_1))$ tyypillisten merkkijonojen kokoelman seuraavsti.

Määritelmä 5.2.1. Olkoon $\beta > 0$. Määritellään *βN-tyypillisten merkkijonojen kokoelmaksi*

$$T_{N\beta} := \left\{ S \in \mathcal{M}^N_{\{0,1\}} : \left| \frac{1}{N} \log_2 \left(\frac{1}{P(S)} \right) - H(X) \right| < \beta \right\}.$$

Edelleen näytimme, more or less, seuraa Lemmasta 4.2.6, että tälle joukolle pätee

$$2^{N(H(X)-\beta)}(1 - \frac{c}{\beta^2 N}) \leq |T_{\beta N}| \leq 2^{N(H(X)+\beta)}$$

ja kaikilla $S \in T_{\beta N}$ pätee

$$2^{-N(H+\beta)} < P(S) < 2^{-N(H-\beta)},$$

missä $H := H(X)$.

Tahtoo siis sanoa, että tyypillisiä merkkijonoja, joissa on noin $p_0 N$ nollaa ja $p_1 N$ ykköstä, on noin 2^{NH} kappaletta, ja $P(T_{N\beta}) \approx 1$. Eli todennäköisyyden mielessä lähes kaikki merkkijonot ovat tyypillisiä, mutta mikäli $H < 1$, niin *lukumääräisesti* hyvin harva merkkijono on tyypillinen. Tähän huomioon nojasi vahavasti shannonin lähdekoodauslause.

Jotta voisimme imitoida kohinaista kirjoituskonetta, huomataan että mikäli olemme lähettäneet kanavalle jonkin bittijonon S, niin vastaanotettua signaalia kuvaa todennäköisyysjakauma $(Y^N|X^N = S)$. Tässä jakaumassa on myöskin tyypillisiä alkioita noin $2^{NH(Y^N|X^N=S)}$ kappaletta. Kun lasketaan, paljonko tällaisen jakauman entropia on *keskimääräisesti*, huomataan että jos kanavalle on lähetetty jokin symboli, niin keskimääräisesti vastaanotettua signaalia kuvaamassa jakaumassa on noin $2^{NH(Y|X)}$ tyypillistä alkiota. Kaikkien mahdollisten vastaanotettujen viestien jakaumassa puolestaan on noin $2^{NH(Y)}$ tyypillistä alkiota. Jos nyt haluamme imitoida kohinaista kirjoituskonetta, ja valita lähetettävistä merkkijonoista sellaisia jotka harvoin kuvautuvat päällekkäin kanavalla, meidän tarvitsee siis *valita sellaisia lähetettäviä merkkijonoja joita vastaavien vastaanotettujen merkkijonojen jakaumien tyypillisten merkkijonojen joukot ovat erillisiä.* Koska tyypillisiä merkkijonoja on vastaanottopäässä yhteensä noin $2^{NH(Y)}$ kappaletta, ja keskimäärin yhden lähetetyn merkkijonon tuottaa $2^{NH(Y|X)}$ tyypillistä merkkijonoa, niin voimme keskimäärin toivoa että voimme valita enintään

$$\frac{2^{NH(Y)}}{2^{NH(Y|X)}} = 2^{N(H(Y)-H(Y|X)} = 2^{NI(X;Y)} \leq 2^{N\,\mathrm{Cap}(Y)}$$

merkkijonoa. Jos onnistumme tällaiset merkkijonot valitsemaan, niin lähetämme N mittaisia blokkeja viestiäksemme $NI(X;Y)$ mittaisia viestejä, eli viestinopeutemme olisi juurikin $I(X;Y)$.

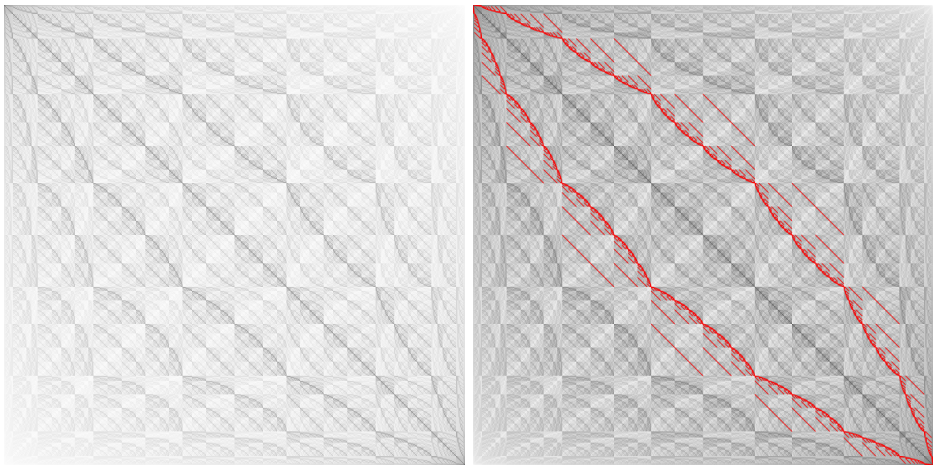
Tämä huomio on Shannonin kanavakoodauslauseen ytimessä. Sen formaalin käsittelyn sivuutamme, mutta mainitsemme että tärkeänä osana on tutkia *yhteistyypillisiä* merkkijonopareja (S,T), jossa S kuvastaa kanavalle lähetettyä merkkijonoa ja T vastaanotettua. Merkkijonopari on yhteistyypillinen jos S ja T ovat tyypillisiä marginaalijakaumissaan (eli niissä on suunnilleen oikea määrä ykkösiä ja nollia) ja pari (S,T) on tyypillinen kanavajakaumasta muodostetun tulojakauman $(X,Y)^N$ alkio (eli merkkijonojen ykköset ja nollat poikkeavat toisistaan noin ε verran ajasta.) Kuvassa 5.8 on korostettu aiemmin tutkitun blokkikanavan tiettyä tyypillistä osajoukkoa.

Miten Shannonin kanavakoodauslause sitten todistetaan? Karkea idea[2]:

1. Valitse virhetaso ε ja kanavanopeus $R < \mathrm{Cap}(Y)$.

2. Ota niin iso N, että kanavajakauma $(X,Y)^N$ muistuttaa 'tarpeeksi' konekirjoitusjakaumaa.

[2]Jälleen kerran, katso tarkka todistus MacKayn kirjasta, [Mac03, Luvut 9-10,].



(a) $N = 12$ (b) $N = 12$, highlight

Kuva 5.8: Kanavajakauman eräs tyypillinen osajoukko korostettuna. Huomaa, että todennäköisimmät alkiot eivät ole tyypillisiä.

3. Valitse satunnaisesti K kappaletta lähetettävää koodia (missä K on s.e. tarvittava viestinopeus voidaan saavuttaan).

4. Laske, että keskimääräisesti tällaisen koodin virhetodennäköisyys on alle ε.

5. Koska keskimääräinen koodin keskimääräinen virhe on alle ε, täytyy välttämättä yhden satunnaisen koodin toteuttaa, että keskimääräinen virhe on alle ε.

6. Tarkalleen ottaen Shannonin lause haluaa maksimivirheen alle ε, joten teet ovelan 'karsitaoperaation' jossa poistat merkkijonoja s.e. myös maksimivirhe pysyy pienenä.

7. Profit.

Shannonin lauseen todistus antaa siis vain olemassaolon tällaiselle koodille, eikä kerro mikä koodi tarkalleen ottaen on kyseessä.

Luku 6

Kaikki loput informaatioteoriaan liittyvät asiat

Lopusksi käydään lävitse hieman aiemmasta materiaalista irtonaisempia aiheita, jotka toimivat, toivon mukaan, eräänlaisena kertauksena siihen mitä tähän asti on tehty. Aiheista ei tule tenttiin kysymyksiä.

6.1 Satunnaisuus

"Onko 4 satunnainen"

Ihan kurssin alussa puhuimme siitä, että vaikka TN-teoriaa voi käyttää uhkapelien ja satunnaisten ilmiöiden tutkimiseen, ei se tarkoita että sitä voisi käyttää ainoastaan siihen. Erityisesti mainitsimme, että TN-teorian 'määritelmässä' voi ihmistä hieman huolettaa että miten 'satunnaisuus' voitaisiin määritellä. Kaksi tapaa lähestyä kysymystä ovat "ennustettavuus" sekä "kuvailtavuus".

Ennustettavuuden ideana on, että *lähde* josta tulee ulos merkkejä on satunnainen, mikäli ensimmäiset N merkkiä havainnoituasi et osaa täyttä arvausta paremmin ennustaa seuraavaa merkkiä. Voidaan siis ajatella, Coxin lauseen ja Bayesin päättelyn hengessä, että lähde on satunnainen jos epävarmuuttasi seuraavalle merkille kuvaa tasajakauma riippumatta siitä että montako merkkiä olet tähän mennessä havainnut.

Kuvailtavuuden ideana on, että merkkijono S on satunnainen mikäli sen Kolmogorov kompleksisuus on vähintään merkkijonon pituus, eli mikäli $K(S) \geq |S|$. Tämä tietenkin tarkoittaa, että meidän pitää puhua merkkijonon satunnaisuudesta jossakin kuvailun kontekstissa, eli kiinnittää jokin Turingin kone (Mystisk box). Tämä satunnaisuuden määritelmä tunnetaan nimellä *Martin-Löf satunnaisuus* tai *algoritminen satunnaisuus.*

Nämä kaksi tapaa kuvailla satunnaisuus liittyvät toisiinsa ja kommunikaatioteoriaan. Kuvitellaan että meillä on lähde, joka tuottaa meille ykkösiä ja nollia, joita haluamme lähettää jonkin kohinattoman kanavan kautta vastaanottajalle. Luodaan tätä varten koodausjärjestelmä jossa on *ennustaja* P, eli jokin järjestelmä/ohjelma/MystiskBox joka yrittää arvata seuraavan lähteestä tulevan merkin. Jokaisessa vaiheessa koodauksessa yritetään ensin arvata P:llä seuraava merkki, ja sitten kanavalle lähetetään nolla mikäli arvaus meni oikein ja ykkönen mikäli arvaus oli väärin. Vastaanottopäässä on täydellinen kopio P:stä, ja viestiä vastaanotettaessa yritetään ensin arvata P:llä seuraava merkki, ja sitten katsotaan kanavalta tulevasta bitistä että menikö arvaus oikein vai väärin. Tämä systeemi kommunikoi täydellisesti lähteen bitit. Tärkeä huomio tässä kuitenkin on se, että mikäli P arvaa seuraavan bitin oikein paremmin kuin puolet ajasta, niin se lähettää maailmalle enemmän nollia kuin ykkösiä. Tämä puolestaan tarkoittaa, että voimme edelleen nopeuttaa kommunikaatiota Huffman-koodauksella tai vastaavalla. Eli viestilähteen ennustettavuus tarkoittaa parempaa kompressiota, eli pienempää Kolmgorov kompleksisuutta. Katso lisää esimerkiksi lähteistä [**?**], [Mac03, Section 6, s. 110–].

6.2 Alkuluvut ja komprsessio

Todistetaan tässä, että alkulukuja on ääretön määrä käyttäen Kolmogorov-kompleksisuutta. Tarvitsemme todistuksessa tietoa, että jokaisen luvun voi esittää alkulukujen tulona, mutta emme esimerkiksi tämän esityksen yksikäsitteisyyttä.

Lemma 6.2.1. *Jokainen luonnollinen luku voidaan esittää alkulukujen tulona.*

Todistus. Tehdään vastaoletus; on olemassa luonnollinen luku jota ei voi esittää alkulukujen tulona. Koska tällainen luku on olemassa, niin joukko

$$A = \{k \in \mathbb{N} \mid \text{ Lukua } k \text{ ei voi esittää alkulukujen tulona.}\}$$

on epätyhjä, jolloinka voimme valita sieltä pienimmän alkion $n = \min A$. Nyt jos luku n ei ole jaollinen muilla luvuilla kuin itsellään tai ykkösellä, on se alkuluku ja siten triviaalisti alkulukujen tulo. On siis välttämättä $n = km$. Nyt jos kumpikin luvuista k ja m voitaisiin esittää alkulukujen tulona, niin niiden tulo n voitaisiin myös. Täten jompaakumpaa näistä luvuista ei voida esittää alkulukujen tulona. Kuitenkin kumpikin luvuista on pienempi kuin n, jonka kuului olla pienin luku jota ei voi esittää alkulukujen tulona. Tämä on ristiriita, joten alkuperäinen väite on tosi. □

Nyt mikäli alkulukuja olisi vain äärellinen määrä, K kpl., voisi minkä tahansa luvun N kirjoittaa tulona $p_1^{N_1} \cdots p_K^{N_K}$. Välttämättä $N_j \leq \log_{N_K}(N)$ kaikilla $j = 1, \ldots, K$. Jokainen luvuista N_j voidaan esittää $\log_2(N_j) \leq \log_2(\log_{p_K}(N))$ bitillä, eli kaikki luvut N voidaan esittää käyttämällä enintään $C \log(\log(N))$ bittiä. Tämä on ristiriita, sillä erilaisia lukuja vähintään N on N kappaletta, ja ilmaisuja joiden pituus on enintään $C \log\log(N)$ on enintään $C' \log(N)$, joten suurilla N saadaan ristiriita.

(Katso lisää esim. `https://en.wikipedia.org/wiki/Euclid%27s_theorem#Proof_using_information_theory`.)

6.3 Informaatioteoria ja Kryptografia

Teemana käsitteet

- Unicity distance
- Englannin entropia (noin 1.3–2.3 bittiä per kirjain)

6.3.1 Landauerin periaate

One of the consequences of the second law of thermodynamics is that a certain amount of energy is necessary to represent information. To record a single bit by changing the state of a system requires an amount of energy no less than kT, where T is the absolute temperature of the system and k is the Boltzman constant. (Stick with me; the physics lesson is almost over.)

Given that k = 1.38x10-16 erg/ Kelvin, and that the ambient temperature of the universe is 3.2 Kelvin, an ideal computer running at 3.2 K would consume 4.4x10-16 ergs every time it set or cleared a bit. To run a computer any colder than the cosmic background radiation would require extra energy to run a heat pump.

Now, the annual energy output of our sun is about 1.21x1041 ergs. This is enough to power about 2.7x1056 single bit changes on our ideal computer; enough state changes to put a 187-bit counter through all its values. If we built a Dyson sphere around the sun and captured all its energy for 32 years, without any loss, we could power a computer to count up to 2192. Of course, it wouldn't have the energy left over to perform any useful calculations with this counter.

But that's just one star, and a measly one at that. A typical supernova releases something like 1051 ergs. (About a hundred times as much energy would be released in the form of neutrinos, but let them go for now.) If all of this energy could be channeled into a single orgy of computation, a 219-bit counter could be cycled through all of its states.

These numbers have nothing to do with the technology of the devices; they are the maximums that thermodynamics will allow. **And they strongly imply that brute-force attacks against 256-bit keys will be infeasible until computers are built from something other than matter and occupy something other than space.**

Bruce Schneier, Applied Cryptography (1996), s. 157

Kts. Feynman: Lectures on Computation sivut x–y. [**?**]

6.4 Kompleksisuus versus sofistikaatio

Katso Aaronson, Carroll ja Ouellette [ACO14].

6.5 Yhden pikselin kamera

Kts. One pixel camera [AMS].

Kirjallisuutta

[AMS] American Mathematical Society *Compressed Sensing MakesEvery Pixel Count* https://www.ams.org/publicoutreach/math-history/hap7-pixel.pdf

[ACO14] S. Aaronson, S. M. Carroll and L. Ouellette *Quantifying the Rise and Fall of Complexity in Closed Systems: The Coffee Automaton* https://arxiv.org/abs/1405.6903

[Jay03] E. T. Jaynes. *Probability theory*. Cambridge University Press, Cambridge, 2003. The logic of science, Edited and with a foreword by G. Larry Bretthorst.

[Mac03] David J. C. MacKay. *Information theory, inference and learning algorithms*. Cambridge University Press, New York, 2003.

[Sch96] B. Schneier Applied Cryptography *Wiley*, 1996

[Sha48] C. E. Shannon. A mathematical theory of communication. *Bell System Tech. J.*, 27:379–423, 623–656, 1948.

[Siv06] D. S. Sivia. *Data analysis*. Oxford Science Publications. Oxford University Press, Oxford, second edition, 2006. A Bayesian tutorial, With J. Skilling.