Vilius Stakėnas

Pusamžė teorija

Dešimtmečių, o kartais ir šimtmečių prireikia, kol pavienės įžvalgos, uždaviniai ir rezultatai, daugybę kartų patikrinti, įrodyti ir interpretuoti susiklosto į savitą tyrimų sritį ir metodus turinčią matematinę teoriją. Bet būna ir išimčių. Viena iš jų – informacijos teorija. Vieningai sutariama ir dėl jos atsiradimo datos ir dėl prioriteto.

„Claude Shannono,*Matematinė ryšio teorija*[1] paskelbta 1948 metų liepos–spalio mėnesiais, yra informacijos amžiaus *Magna Carta*. Shannono atrasti fundamentalūs duomenų spūdos ir perdavimo dėsniai reiškė informacijos teorijos gimimą", – rašoma apžvalgoje, skirtoje informacijos teorijos 50-mečiui.[2]

Visas didelio formato kelių šimtų puslapių žurnalo *IEEE Transactions on Information Theory* numeris skirtas įvairių informacijos teorijos sričių 50 metų raidos apžvalgoms. Jų daug – net 25. Ir daugelis prasideda panašiai: „Po to, kai 1948 metais C. Shannonas..." Šiame straipsnyje aptariamos tik kelios pagrindinės informacijos teorijos temos, kurioms pradžią davė jau minėtas C. Shannono darbas.

Informacijos šaltiniai ir matematinis požiūris į juos

Informacijos šaltinių būna kuo įvairiausių – knygos, laikraščiai, televizija ir gandai... Apie visus juos iš karto galima kalbėti tik būnant nuosekliu Platono šalininku. Kaip Platonui, tarkime, visi medžiai yra grynosios medžio idėjos atspindžiai, taip ir matematikams svarbiausi yra jų sukurti matematiniai modeliai, kuriuos realūs reiškiniai geriau ar blogiau atitinka.

Tad koks gi turi būti informacijos šaltinio matematinis modelis? Kol kas kliaukimės tik intuicija. Informaciją suteikia įvykiai. Mestas į viršų kamuolys nukrito ant žemės. Šis įvykis mums nesuteikė jokios informacijos, nes taip visada būna. Šiąnakt lijo. Šį teiginį suprantame kaip informatyvų – juk galėjo ir nelyti. Taigi informaciją suteikia atsitiktiniai įvykiai. Visai natūraliai prieiname prie tokios pradinės informacijos šaltinio sampratos:

- *informacijos šaltinis yra procesas, kurio baigtis yra atsitiktinė.*

Ta galimų baigčių aibė gali būti tiek baigtinė, tiek begalinė, tačiau apsiribokime baigtinėmis aibėmis – tai labiau atitinka tiek mūsų prigimtį, tiek veiklos pobūdį.

[1] Shannon C. E., The mathematical theory of communication, *Bell Syst. Techn. J.*, vol 27, July 1948 p. 379–423; Oct. 1948, p. 623–656.

[2] Verdú S., Fifty Years of Shannon Theory, *IEEE Transactions on Information Theory*, vol. 44, N. 6, 1998, p. 2057–278.

Mes šiek tiek arčiau priartėsime prie informacijos įprastinės sampratos, jeigu galimas baigtis žymėsime kokios nors abėcėlės raidėmis. Pagaliau matematiniam tyrinėjimui visai nesvarbu, koks tas eksperimentas, kurio baigtys yra abėcėlės raidės. Svarbu, kad jos mums atsitiktinai pasirodo. Taigi prieiname prie tokio visai „sauso“ informacijos šaltinio apibrėžimo:

- *informacijos šaltinis yra atsitiktinis dydis X, įgyjantis reikšmes iš baigtinės aibės $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$ (abėcėlės).*

Nors tai šiek tiek juokinga, bet į besiartinantį pašnekovą galime žvelgti kaip į atsitiktinį dydį, kuris tuoj tuoj įgys reikšmę.

O dabar išmokime matuoti informacijos kiekį, kurį suteikia šaltinio perduotas simbolis. Jei pašnekovas pradės pokalbį žodžiu „labas“, tai mūsų nenustebins, tačiau žodis „sudie“ suteiks daug peno apmąstymams apie jo būklę. Taigi simboliai perduoda skirtingą informacijos kiekį.

Panagrinėkime tokį paprastą pavyzdį. Tarkime, „Žalgirio“ futbolo klubas susitinka su Mančesterio „United“, o mes, pasibaigus rungtynėms, privalome parašyti ataskaitą, kurioje reikia išanalizuoti rungtynių baigties priežastis. Objektyviai „United“ yra stipresnis futbolo klubas, tad jei jis nugalės, niekas per daug nenustebs. Ir mūsų ataskaita nebus didelė, parašysime keletą visiems žinomų teiginių ir viskas. Tačiau jeigu „Žalgiris“ pasiektų lygiąsias, tikriausiai analizuodami surastume daug įdomių dalykų. Galbūt „Žalgirio“ klube sužibo nauja žvaigždė, o gal „United“ klube reikalai pašlijo. O jeigu „Žalgiris“ laimėtų?

Taigi jau galime formuluoti išvadą:

- *atsitiktinė baigtis suteikia tuo daugiau informacijos, kuo mažesnė tikimybė, kad ji pasirodys.*

Grįškime prie formalios informacijos šaltinio sąvokos. Tarkime, šaltinis X gali perduoti abėcėlės $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$ simbolius su tikimybėmis $p_1, p_2, \ldots, p_n$. Tarsime, kad visos tikimybės teigiamos: $p_i \neq 0$. Užrašas $P(X = a_i) = p_i$ reiškia, kad simbolio a_i pasirodymo tikimybė yra p_i. Pabandysime nustatyti, kaip reiktų skaičiuoti simbolio a_i perduodamą informacijos kiekį $I(a_i)$. Aišku, kad informacijos kiekis nei padidės, nei sumažės, jeigu raidę a_i pakeisime kokia nors kita raide. Taigi informacijos kiekis priklauso ne nuo to, kokios abėcėlės raidės yra naudojamos, bet nuo pačių tikimybių. Žengėme dar vieną žingsnelį aiškindamiesi, ko gi iš tiesų norime. Informacijos kiekiui matuoti reikia sukonstruoti funkciją $f : (0, 1] \to [0, \infty)$. Tada simbolio, kurio pasirodymo tikimybė lygi p, suteikiamą informacijos kiekį reikšime dydžiu

$$I(a) = f(p).$$

Tačiau kokias funkcijas šiam tikslui pasitelkti? Vienas reikalavimas beveik nekelia abejonių:

- *$f(u)$ turi būti tolydi, griežtai monotoniškai mažėjanti funkcija,* $f(1) = 0$.

Sąlyga $f(1) = 0$ reiškia, kad, mūsų manymu, simbolis, pasirodantis su tikimybe 1, t. y. visada, nesuteikia jokios informacijos.

Norėdami suformuluoti kitą sąlygą, įsivaizduokime, kad simbolis a sudarytas iš dviejų dalių: $a = a_1 a_2$. Iš pradžių pasirodo raidė a_1, o po to – nepriklausomai nuo pirmosios ir antroji raidė a_2. Tarkime, kad raidės a_1 pasirodymo tikimybė yra p, tada ji mums perduoda informacijos kiekį $f(p)$. Jeigu raidės a_2 pasirodymo tikimybė lygi q, tai ji perduoda informacijos kiekį, lygų $f(q)$. Taigi iš viso simbolio $a = a_1 a_2$ pasirodymas suteikia $f(p) + f(q)$ informacijos. Kadangi a pasirodymo tikimybė lygi pq, tai turi būti:

- $f(pq) = f(p) + f(q)$ su visais $p, q \in (0, 1]$.

Pasirodo, kad funkcijų, tenkinančių abi sąlygas, nėra tiek daug. Tai yra logaritmų šeima

$$f_d(p) = \log_d \frac{1}{p}, \quad d > 1.$$

Galime pasirinkti vieną iš jų. Geriausiai tinka logaritmas, kurio pagrindas $d = 2$. Taigi nuo šiol informacijos kiekį, kurį suteikia simbolis, pasirodantis su tikimybe $p > 0$, matuosime dydžiu

$$I(p) = \log_2 \frac{1}{p}.$$

Jei simbolis pasirodo su tikimybe $1/2$, tai jo pasirodymas suteikia vieną informacijos vienetą, kurį vadinsime *bitu*. Taigi metus simetrišką monetą, tiek skaičiaus, tiek herbo pasirodymas suteikia po bitą informacijos.

Galime apibrėžti ir viso informacijos šaltinio X, perduodančio simbolius $a_1, a_2, \ldots, a_n$ su tikimybėmis $p_1, p_2, \ldots, p_n$, informatyvumą. Jį reikšime dydžiu

$$H(X) = p_1 \log_2 \frac{1}{p_1} + p_2 \log_2 \frac{1}{p_2} + \ldots + p_n \log_2 \frac{1}{p_n}.$$

Šis svarbus dydis vadinamas *šaltinio entropija*. Tai tiesiog atskirų simbolių suteikiamų informacijos kiekių matematinis vidurkis.

„Tikrieji“ informacijos šaltiniai perduoda anaiptol ne po vieną raidę. Tačiau jau visai nesunku priartinti mūsų „matematinio“ šaltinio sąvoką prie „tikrųjų“ šaltinių. Informacijos šaltiniu, perduodančiu n ilgio simbolių sekas, vadinsime atsitiktinių dydžių seką

$$X_1, X_2, \ldots, X_n;$$

čia kiekvienas dydis įgyja reikšmes iš abėcėlės $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$. Galime nagrinėti netgi begalines simbolių sekas perduodančius šaltinius.

Realioje kalboje kiekvienas kitas garsas (arba, jei kalba užrašyta, raidė) nėra nepriklausomas nuo ką tik ištartojo. Juk jeigu ištartas garsas „v“, tai mažai tikėtina, kad kitas bus „f“. Tačiau visiškos priklausomybės taip pat nėra. Kitaip iš anksto žinotume visa, kas bus pasakyta (kartais taip ir būna). Paprasčiausias šaltinio atvejis – kai visi dydžiai yra nepriklausomi. Tokį šaltinį vadinsime šaltiniu be atminties. Suprantama, realūs šaltiniai nėra šaltiniai be atminties, tačiau priklausomybė yra tuo mažesnė, kuo simboliai garsų ar raidžių eilėje yra toliau vienas nuo kito.

Didžiųjų skaičių dėsnis ir kodavimas

Kiekvienas informacijos šaltinis turi savų bruožų, pavyzdžiui, abėcėlę. Kita vertus, informacijos šaltinio generuota simbolių seka turi būti tam tikru būdu užrašyta, kad fiziniai perdavimo kanalai galėtų ją perduoti. Skaitmeniniams kanalams būtina, kad viskas būtų užrašyta dvinarės abėcėlės $\mathcal{B} = \{0, 1\}$ simboliais.

Taigi akivaizdi būtinybė vienos abėcėlės simbolių srautą keisti kitos abėcėlės simbolių srautu, t. y. koduoti. Aptarkime formalias tokio kodavimo sąlygas labai paprastu atveju. Informacijos šaltinis perduoda mums abėcėlės $\mathcal{A} = \{a_1, a_2, \ldots, a_L\}$ simbolių srautą, kurį skaidome N ilgio blokais (žodžiais), o pastaruosius būtina keisti vienodo ilgio abėcėlės $\mathcal{B} = \{b_1, b_2, \ldots, b_K\}$ simbolių blokais (žodžiais). Kokio ilgio $\mathcal{B}$ abėcėlės žodžius reikia naudoti? Skirtingų $\mathcal{A}$ abėcėlės N ilgio žodžių yra L^N; jei naudosime M ilgio $\mathcal{B}$ žodžius, jų turėsime K^M. Kad jų užtektų visiems galimiems $\mathcal{A}$ žodžiams koduoti, turi būti patenkinta sąlyga

$$K^M \geq L^N, \qquad \text{arba} \qquad \frac{M}{N} \geq \frac{\log_2 L}{\log_2 K}. \tag{1}$$

Santykis M/N rodo, kiek $\mathcal{B}$ abėcėlės simbolių tenka vienam $\mathcal{A}$ abėcėlės simboliui koduoti.

Tarkime, kad $\mathcal{B}$ abėcėlės žodžius perduoda koks nors fizinis kanalas, turintis jam būdingą perdavimo greitį, kurio viršyti neįmanoma. Dėl paprastumo tarkime, kad vieną $\mathcal{B}$ abėcėlės simbolį kanalas perduoda per vieną sąlyginį laiko vienetą (pavyzdžiui, mili-, mikro- sekundę). Tačiau mums rūpi abėcėlės $\mathcal{A}$ simbolių perdavimo greitis. Kadangi N ilgio $\mathcal{A}$ žodžiui perduoti kanalas sugaišta M laiko vienetų, tai vienam simboliui perduoti sugaiš $\varrho = M/N$. Pavadinkime šį dydį *perdavimo sąnaudomis* (atvirkštinį dydį, kuris reiškia per vieną laiko vienetą perduodamų $\mathcal{A}$ simbolių skaičių, galėtume pavadinti *perdavimo greičiu*). Aišku, kad perdavimo sąnaudos priklauso nuo to, kaip $\mathcal{A}$ žodžius koduojame $\mathcal{B}$ abėcėlės žodžiais. Tačiau iš (1) sąryšio matome, kad perdavimo sąnaudoms teisingas įvertis

$$\varrho \geq \frac{\log_2 L}{\log_2 K}.$$

Pavyzdžiui, jeigu $\mathcal{A}$ yra lietuvių kalbos abėcėlė ($L = 32$), o $\mathcal{B} = \{0, 1\}$, tai perdavimo sąnaudos negali būti mažesnės už

$$\frac{\log_2 32}{\log_2 2} = 5$$

laiko vienetus vienam simboliui. Nejau neįmanoma paspartinti perdavimo? Išvesdami (1) sąryšį buvome nepaprastai pedantiški: rūpinomės, kad visiems N ilgio $\mathcal{A}$ abėcėlės žodžiams užtektų $\mathcal{B}$ abėcėlės žodžių. Jei $\mathcal{A}$ yra lietuvių kalbos abėcėlė, tai darėme prielaidą, kad, pavyzdžiui, gali tekti koduoti ir tokius žodžius: žžžghž..... rrttžž. Galbūt tokie žodžiai ir gali pasitaikyti kokiame nors keistame literatūriniame tekste.

O dabar apie didžiųjų skaičių dėsnį. Tiesą sakant, visi jį žino ir juo pasikliauja. Norėdamas objektyviau įvertinti moksleivio žinias, mokytojas išveda aritmetinį pažymių vidurkį. Ir moksleivis, ir mokytojas neabejoja, kad aritmetinis vidurkis objektyviau atspindi žinias ir sugebejimus negu bet kuris vienas pažymys. Kodėl?

Panagrinėkime šią situaciją „grynesniu“ pavidalu. Atlikime mintinį eksperimentą: įsivaizduokime, kad didelis skaičius lošėjų (tarkime, $N \approx 1000$) po didelį skaičių kartų (tarkime, $n \approx 100.000$) meta po simetrišką lošimo kauliuką, o po to skaičiuoja aritmetinį iškritusių akučių vidurkį. Mūsų praktinė nuojauta sako, kad daugumai lošėjų akučių $1, 2, 3, 4, 5, 6$ pasirodymų skaičiai nedaug skirsis:

$$n_1 \approx n_2 \approx n_3 \approx n_4 \approx n_5 \approx n_6 \approx \frac{n}{6}.$$

Tada žymėdami X_k k-uoju metimu iškritusių akučių skaičių, dauguma lošėjų gaus tokį rezultatą:

$$\frac{X_1 + X_2 + \ldots + X_n}{n} = 1 \cdot \frac{n_1}{n} + 2 \cdot \frac{n_2}{n} + \ldots + 6 \cdot \frac{n_6}{n} \approx 3,5.$$

Jeigu kiekvienam lošėjui bus duota po vienodą nesimetrišką kauliuką, kurį mėtant akutės $1, 2, \ldots, 6$ krinta su tikimybėmis $p_1, p_2, \ldots, p_6$, tai dauguma lošėjų gaus tokį rezultatą:

$$\frac{X_1 + X_2 + \ldots + X_n}{n} \approx 1 \cdot p_1 + 2 \cdot p_2 + \ldots 6 \cdot p_6.$$

Šis tvirtinimas ir yra didžiųjų skaičių dėsnis. Pabandykime jį išreikšti griežtesne matematine kalba.

Didžiųjų skaičių dėsnis. *Tegu atlikta n nepriklausomų to paties atsitiktinio dydžio, įgyjančio skaitines reikšmes $x_1, x_2, \ldots, x_n$ su tikimybėmis $p_1, p_2, \ldots, p_n$, stebėjimų. Jeigu $X_1, X_2, \ldots, X_n$ yra šiuose stebėjimuose gautos reikšmės, tai*

$$P\left(\frac{X_1 + X_2 + \ldots + X_n}{n} \approx MX\right) \approx 1; \qquad (2)$$

čia $MX = x_1p_1 + x_2p_2 + \ldots + x_np_n$.

Nors tokia formuluotė pakankamai gerai perteikia didžiųjų skaičių dėsnio esmę, tačiau matematiniu požiūriu turi didelių trūkumų. Juk simbolio $\approx$ reikšmė gana miglota. Suteikę jam griežtą prasmę, gausime tokią didžiųjų skaičių dėsnio formuluotę.

Didžiųjų skaičių dėsnis. *Tegu atliekama n nepriklausomų to paties atsitiktinio dydžio, įgyjančio skaitines reikšmes $x_1, x_2, \ldots, x_n$ su tikimybėmis $p_1, p_2, \ldots, p_n$, stebėjimų. Tegu $X_1, X_2, \ldots, X_n$ yra šiuose stebėjimuose gautos reikšmės. Tada bet kokiems teigiamiems skaičiams ε, δ egzistuoja toks skaičius $n_0 = n_0(\varepsilon, \delta)$, kad*

$$P\left(\left|\frac{X_1 + X_2 + \ldots + X_n}{n} - MX\right| \le \varepsilon\right) \ge 1 - \delta,$$

kai $n \ge n_0$, čia $MX = x_1p_1 + x_2p_2 + \ldots + x_np_n$.

Didžiųjų skaičių dėsnį pirmasis suformulavo Jakobas Bernoulli (1654–1705), žymiosios matematikų Bernoulli dinastijos pradininkas. Pirmoji užuomina apie didžiųjų skaičių dėsnį Bernoulli matematiniame dienoraštyje labai paprasta ir aiški: „Aš mažiau nukrypstu nuo tikrojo santykio, kai stebiu daugiau kartų“. Dėsnis suformuluotas pagrindiniame Jakobo Bernoulli'o tikimybių teorijos veikale „*Ars Conjectandi*“ (Spėjimo menas), kuriuo, galima sakyti, prasidėjo tikroji tikimybių teorijos raida. Šį veikalą, praėjus aštuoniems metams po autoriaus mirties, išleido jo sūnėnas Nicolas Bernoulli. Jakobui Bernoulli priklauso ir daugelio kitų matematinių atradimų autorystė. Jis ypač žavėjosi jo paties tyrinėta logaritmine spirale. Ši kreivė su užrašu *Eadem mutata resurgo* (pasikeitusi atgimstu tokia pat) iškalta jo antkapyje. Panašiai galima būtų pasakyti ir apie didžiųjų skaičių dėsnį, kuris įvairiais pavidalais, bet nepasikeitusia prasme pasirodo įvairiuose matematiniuose kontekstuose.

Kuo gi didžiųjų skaičių dėsnis gali padėti kodavimui? Tarkime, kad šaltinis, kurio informaciją norime koduoti, neturi atminties, t. y. vienas simbolis nedaro įtakos vėliau perduodamų simbolių pasirodymui. Tokį šaltinį aprašysime nepriklausomų atsitiktinių dydžių seka

$$X_1, X_2, X_3, \ldots;$$

čia X_n įgyja reikšmes iš abėcėlės $\mathcal{A} = \{a_1, a_2, \ldots, a_L\}$. Be to, tarkime, kad tikimybė $P(X_i = a_k) = p(a_k)$ priklauso tik nuo a_k, t. y. visi X_i įgyja reikšmes iš $\mathcal{A}$ su vienodomis tikimybėmis. Tarkime, šaltinio perduodamą simbolių srautą stebime tol, kol perduodamas N ilgio žodis. Jeigu gavome žodį $x = x_1x_2\ldots x_n$, tai jo perduotas informacijos kiekis lygus

$$I(x) = I(x_1) + I(x_2) + \ldots + I(x_n).$$

Kadangi informacijos kiekis priklauso tik nuo tikimybių, tai

$$I(x) = I(p(x_1)) + I(p(x_2)) + \ldots + I(p(x_n)), \quad I(p(x_k)) = \log_2 \frac{1}{p(a_k)}.$$

Žodis x yra atsitiktinis, sudarytas iš n nepriklausomų komponenčių, taigi šią lygybę galime interpretuoti kaip skaitines reikšmes įgyjančio atsitiktinio dydžio n nepriklausomų stebėjimų rezultatų sumą. Tada pažymėję

$$Y_k = I(p(x_k)) = \log_2 \frac{1}{p(x_k)},$$

šiems dydžiams (2) didžiųjų skaičių dėsnį galime užrašyti taip:

$$P\Big(\frac{I(x)}{n} = \frac{Y_1 + Y_2 + \ldots + Y_n}{n} \approx p(a_1)\log_2\frac{1}{p(a_1)} + \ldots + p(a_n)\log_2\frac{1}{p(a_n)}\Big) \approx 1.$$

Tačiau

$$H = p(a_1)\log_2\frac{1}{p(a_1)} + \ldots + p(a_n)\log_2\frac{1}{p(a_n)}$$

yra šaltinio entropija. Taigi

$$P\Big(\frac{I(x)}{n} \approx H\Big) \approx 1.$$

Gavome, kad su artima vienetui tikimybe šaltinis perduos n ilgio žodį, kurio informacijos kiekis

$$I(x) = I(p(x)) = \log_2\frac{1}{p(x)} \approx nH; \tag{3}$$

čia $p(x)$ yra žodžio x pasirodymo tikimybė. Tokius žodžius vadinsime *tipiniais*, likusieji pasirodo retai, taigi kyla noras jų visai nepaisyti. Kiek yra tipinių žodžių? Kadangi kiekvieno tipinio žodžio x pasirodymo tikimybei iš (3) gauname sąryšį

$$p(x) \approx 2^{-nH},$$

o tipinių žodžių tikimybių suma artima vienetui, tai tipinių žodžių kiekiui T gauname sąryšį

$$1 \approx \sum_{x-tipinis} p(x) \approx T2^{-nH}, \quad T \approx 2^{nH}.$$

Pasirėmę gautomis išvadomis, suformuluokime naują šaltinio perduodamo informacijos srauto kodavimo strategiją:

- pasirinkime N pakankamai didelį, informacijos srautą skaidykime į N ilgio blokus (žodžius);
- į pasitaikančius netipinius žodžius nekreipkime dėmesio, jų pasirodymo tikimybė labai nedidelė, jei n parinkome pakankamai didelį;
- tipinius žodžius, jų yra $\approx 2^{nH}$, koduokime to paties ilgio M abėcėlės $\mathcal{B}$ žodžiais.

Kokio ilgio abėcėlės $\mathcal{B}$ žodžius reikia naudoti, ir kiek $\mathcal{B}$ simbolių reikės vienam $\mathcal{A}$ simboliui užkoduoti? Nuo to priklauso perdavimo greitis.

Jeigu naudosime M ilgio $\mathcal{B}$ žodžius, tai kad jų užtektų visiems tipiniams žodžiams, turi būti

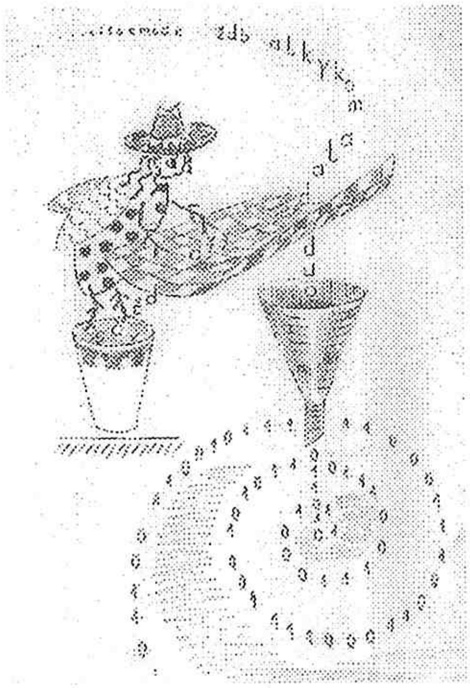
$$K^M \approx 2^{NH};$$

čia K – abėcėlės $\mathcal{B}$ simbolių skaičius. Taigi vienam $\mathcal{A}$ simboliui koduoti sunaudojama

$$\frac{M}{N} \approx \frac{H}{\log_2 K}$$

abėcėlės $\mathcal{B}$ simbolių. Ši lygybė atskleidžia dar vieną požiūrį į entropiją: entropija lygi vienam abėcėlės $\mathcal{A}$ simboliui koduoti sunaudojamų dvinarės abėcėlės $\mathcal{B} = \{0, 1\}$ simbolių skaičiui, kai koduojami tik tipiniai žodžiai.

Jeigu lietuviškai kalbantį šaltinį interpretuotume kaip šaltinį be atminties ir apskaičiuotume jo entropiją, gautume $H \approx 2,6$. Taigi koduojant pakankamai ilgus tipinius fragmentus ir nekreipiant dėmesio į netipinius, vienam simboliui sunaudojama maždaug 2,6 dvinarės abėcėlės simbolių. Lyginant su „kodavimu be nuostolių", perdavimo greitis padidėja beveik dvigubai. Sužinojęs tokią galimybę, skaitytojas gali paklausti, kaip konkrečiai šį efektą pasiekti. Kaip dažnai daro teorinių mokslų atstovai, atsakysime: tai jau visiškai kitas klausimas!



Koduojami tik tipiniai žodžiai!

Nevienodo ilgio žodžių kodai

Ankstesniame skyrelyje aptarėme galimybę pagreitinti informacijos perdavimą koduojant tik „tipinius" šaltinio perduodamus žodžius. Šitaip elgiantis, užtenka trumpesnių iš abėcėlės $\mathcal{B}$ simbolių sudarytų kodo žodžių. Netipinių, retai pasitaikančių žodžių išvis nekoduojame. Taigi kartais šaltinio informaciją ignoruojame, kai jis, mūsų manymu, „nusišneka". Gali kilti mintis, kad ir į juos galėtume kaip nors reaguoti, pavyzdžiui, koduoti ilgesniais žodžiais nei tipinius kodo žodžius. Šitaip prieiname prie nevienodo ilgio žodžių kodo idėjos.

Ši idėja kur kas senesnė už informacijos teoriją. Tekstams perduoti savo sukurtu aparatu S. Morzė sukūrė ir kodą, kurio žodžiai sudaryti iš pasikartojančių dviejų simbolių: taško ir brūkšnio.

Morzės kodas

A	· –	H	····	O	– – –	V	··· –
B	– ···	I	··	P	· – – ·	W	· – –
C	– · – ·	J	· – – –	Q	– – · –	X	– ·· –
D	– ··	K	– · –	R	· – ·	Y	– · – –
E	·	L	· – ··	S	···	Z	– – ··
F	·· – ·	M	– –	T	–		
G	– – ·	N	– ·	U	·· –		

Nesunku suprasti, kuo S. Morzė rėmėsi sudarydamas savo kodą. Dažniau pasitaikančios raidės koduojamos trumpesnėmis, rečiau – ilgesnėmis sekomis. Akivaizdu, kad tai paspartina operatoriaus darbą.

Grįškime prie mūsų matematinio informacijos šaltinio. Tarkime, atsitiktiniai dydžiai

$$X_1, X_2, \ldots$$

aprašo neturintį atminties šaltinį, t. y. X_i nepriklausomai vienas nuo kito įgyja reikšmes iš abėcėlės $\mathcal{A} = \{a_1, \ldots, a_L\}$ su tomis pačiomis tikimybėmis

$$P(X_i = a_k) = p_k.$$

Tarkime, kiekvienam simboliui koduoti iš abėcėlės $\mathcal{B} = \{b_1, \ldots, b_K\}$ simbolių sudarėme po žodį

$$a_k \to c(a_k), \quad c(a_k) = b(k,1)b(k,2)\ldots b(k,n_k)$$

ir informacijos srautą koduojame simbolis po simbolio

$$a_{j_1} a_{j_2} \ldots \to c(a_{j_1})c(a_{j_2})\ldots,$$

tiesiog sujungdami $\mathcal{A}$ simbolių kodo žodžius į vientisą seką.

Kiek vidutiniškai abėcėlės $\mathcal{B}$ simbolių panaudojome vienam $\mathcal{A}$ simboliui koduoti? Jeigu kodavome pakankamai ilgą $\mathcal{A}$ simbolių seką, tarkime, sudarytą iš N simbolių, tai $a_1, \ldots, a_L$ šioje sekoje pasitaikė tikriausiai maždaug

$$N_1 \approx p_1 N, \quad N_2 \approx p_2 N, \quad \ldots, \quad N_L \approx p_L N_L$$

kartų. Žymėdami kodo žodžio $c(a)$ ilgį $|c(a)|$, rasime, kad iš viso sunaudota

$$N_1|c(a_1)| + N_2|c(a_2)| + \ldots + N_L|c(a_L)| \approx N(p_1|c(a_1)| + p_2|c(a_2)| + \ldots + p_L|c(a_L)|)$$

abėcėlės $\mathcal{B}$ simbolių, taigi vienam abėcėlės $\mathcal{A}$ simboliui tenka maždaug

$$\nu_c = |c(a_1)|p_1 + |c(a_2)|p_2 + \ldots + |c(a_L)|p_L$$

$\mathcal{B}$ simbolių. Radome naudojamo kodo c efektyvumo matą. Pageidautina, kad sąnaudas vienam $\mathcal{A}$ simboliui apibūdinantis dydis ν_c būtų kuo mažesnis.

Koduojant nevienodo ilgio žodžiais šaltinio perduodamus simbolius, iškyla štai koks klausimas. Ar tikrai įmanoma pagal kodą atkurti pradinę šaltinio informaciją? Pavyzdys padės mums suprasti problemą.

$\mathcal{A}$	C_1	C_2	C_3	C_4
A	0	0	0	0
B	0	1	10	01
C	1	00	110	011
D	10	11	111	0111

Lentelėje keturiems abėcėlės $\mathcal{A}$ simboliams koduoti nurodyti keturi dvejetainiai kodai. Iškart matyti, kad C_1 kodas niekam tikęs. Antrasis irgi ne geresnis: jei gavome 00, tai nežinia, kurį iš atvejų

$$\mathbf{AA} \longmapsto 00, \quad \mathbf{C} \longmapsto 00$$

jis atitinka. Kodai C_3, C_4 yra geri: jais užkoduotą pirminę informaciją visada galima atkurti. Tačiau jie skiriasi vienu subtiliu bruožu. Norėdami tą skirtumą surasti, nustatykite, kokia pradinė informacija turi būti daugtaškių vietoje:

$$\ldots \overset{C_3}{\longmapsto} 101100, \quad \ldots \overset{C_4}{\longmapsto} 0111010.$$

Kodus, kuriais koduotą informaciją galima atkurti, vadinsime iššifruojamaisiais. Taigi C_3, C_4 – iššifruojami kodai.

Kodu C_3 koduotą simbolį galima atkurti vos tik gavus jį atitinkantį kodo žodį, o kodu C_4 – ne. Šią kodo C_3 savybę lemia tai, kad joks C_3 žodis nėra kito šio kodo žodžio pradžia (priešdėlis). Tokius kodus vadinsime p kodais. Taigi geriausia kodavimui naudoti p kodus. Tik ar visada juos galime sudaryti tokius, kokių norime?

Pavyzdžiui, ar galime sudaryti p kodą 5 simbolių abėcėlei koduoti dvejetainės abėcėlės $\mathcal{B} = \{0, 1\}$ žodžiais, kad jų ilgiai būtų $2, 2, 4, 4, 5$? Atsakymą 1948 metais suformulavo MIT[3] studentas L. Kraftas savo magistro darbe.

Krafto teorema. *Tegu $\mathcal{B}$ yra abėcėlė, turinti K simbolių, o $n_1, \ldots, n_s$ – natūralieji skaičiai. Abėcėlės $\mathcal{B}$ žodžių p kodas $x_1, x_2, \ldots, x_s$, $|x_i| = n_i$, egzistuoja tada ir tik tada, kai teisinga nelygybė*

$$K^{-n_1} + K^{-n_2} + \ldots + K^{-n_s} \leq 1.$$

[3] Massachusetts Institute of Technology.

Kadangi

$$2^{-2}+2^{-2}+2^{-4}+2^{-4}+2^{-5}=\frac{19}{32}<1,$$

tai atsakymas į anksčiau iškeltą klausimą teigiamas.

Jau išsiaiškinome, kad šaltinio, kurio simboliai $a_1,\ldots,a_L$ pasirodo su tikimybėmis $p(a_1),\ldots,p(a_L)$, kodavimui geresnis tas kodas $c(a_1),\ldots,c(a_L)$, kuris turi mažesnį vidutinį žodžio ilgį

$$\nu_c=|c(a_1)|p_1+|c(a_2)|p_2+\ldots+|c(a_L)|p_L.$$

Kokias reikšmes gali įgyti šis dydis? Atsakymas glūdi kitoje Shannono teoremoje.

Shannono teorema. *Kiekvienam p kodui, sudarytam iš abėcėlės $\mathcal{B}=\{b_1,b_2,\ldots,b_K\}$ žodžių, teisinga nelygybė*

$$\nu_c\geq\frac{H}{\log_2 K};$$

čia

$$H=p_1\log_2\frac{1}{p_1}+p_2\log_2\frac{1}{p_2}+\ldots+p_L\log_2\frac{1}{p_L}$$

yra šaltinio entropija. Kita vertus, egzistuoja p kodas, kuriam teisinga nelygybė

$$\nu_c\leq\frac{H}{\log_2 K}+1.$$

Taigi visada egzistuoja toks kodas, kuris vienam šaltinio simboliui koduoti naudoja ne daugiau kaip $1+H/\log_2 K$ abėcėlės $\mathcal{B}$ simbolių. Iš tikrųjų šaltinio simbolių tikimybėms $p_1,\ldots,p_L$ apibrėžkime natūraliuosius skaičius $n_1,n_2,\ldots,n_L$, kad galiotų nelygybės

$$K^{-n_i}\leq p_i<K^{-n_i+1};\qquad(4)$$

čia K yra abėcėlės $\mathcal{B}$ simbolių skaičius. Kadangi

$$K^{-n_1}+K^{-n_2}+\ldots+K^{-n_L}\leq p_1+p_2+\ldots+p_L=1,$$

tai iš Krafto teoremos gauname, kad p kodas c su ilgio $n_1,\ldots,n_L$ žodžiais egzistuoja. Iš (4) išplaukia nelygybė

$$n_i\leq\frac{1}{\log_2 K}\cdot\log_2\frac{1}{p_i}+1,$$

tai kodo vidutiniam žodžio ilgiui teisingas įvertis

$$\nu_c=n_1p_1+n_2p_2+\ldots+n_Lp_L\leq\frac{1}{\log_2 K}\sum_i p_i\log\frac{1}{p_i}+\sum_i p_i=\frac{H}{\log_2 K}+1.$$

••• $\alpha+\omega$ •••

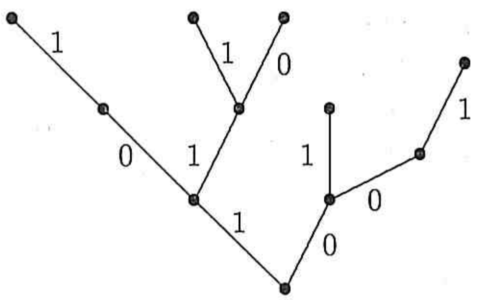
Šis kodas vadinamas Shannono–Fano kodu. Tačiau ar toks kodas tikrai pats geriausias? Ne. Tuo galite įsitikinti suradę Shannono–Fano kodo žodžių ilgius šaltiniui, su tikimybėmis $p_1 = 2^{-k}, p_2 = 1 - 2^{-k}$ perduodančiam abėcėlės $\mathcal{A} = \{1, 2\}$ simbolius, koduoti, kai kodo abėcėlė dvinarė: $\mathcal{B} = \{0, 1\}$. Akivaizdu, kad optimalus kodas yra toks: $c(1) = 0, c(2) = 1$. Shannono–Fano kodas daug blogesnis.

Tad kaip sudaryti patį geriausią kodą? Šįkart nebandysime išsisukti nuo atsakymo.

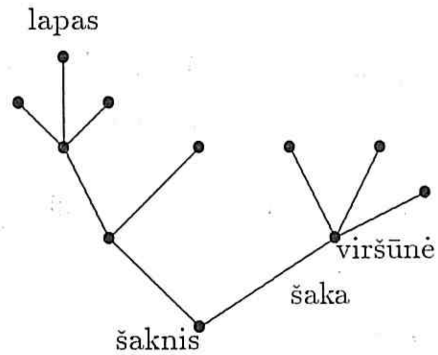
Optimalus kodas

Įsivaizduokime šaltinį, perduodantį savo abėcėlės simbolius su tikimybėmis $p_1, p_2, \ldots, p_n$. Šiuos simbolius mums reikia koduoti dvinarės abėcėlės $\mathcal{B} = \{0, 1\}$ žodžiais, kad sudaryto kodo vidutinis žodžių ilgis būtų kiek galima mažesnis. Pats geriausias šiuo požiūriu (optimalus) kodas egzistuoja. Jeigu žinotume tik tai, iš to naudos būtų ne kažin kiek. Svarbu turėti taisykles (algoritmą), kuriomis vadovaudamiesi visada galėtume šį kodą sudaryti. Tokį algoritmą sugalvojo kitas MIT studentas D. A. Huffmanas, o kodo idėja jam kilo besprendžiant informacijos teorijos namų užduotį.

Šį algoritmą patogiausia paaiškinti naudojantis kodų ir medžių ryšiu. Žinoma, naudosime ne tikrus, bet matematinius medžius, tokius, kaip pavaizduota 1 brėžinyje.



1 brėž.



2 brėž.

Taigi mūsų medis yra grafas, turintis vienintelę apatinę viršūnę, kurią pavadinsime „šaknimi“. Iš viršutinių viršūnių neišeina jokios šakos, jas mes vadinsime „lapais“. Pastebėkime, kad iš kiekvienos viršūnės išeina ne daugiau kaip 2 šakos, o iš šaknies į kiekvieną lapą veda vienintelis kelias. 1 brėžinio medį vadinsime dvinariu medžiu, nes iš jokios viršūnės neišeina daugiau kaip 2 šakos. Medį, kuris turi viršūnių su r šakomis, bet ne daugiau, vadinsime r-nariu medžiu. Pavyzdžiui, 2 brėžinyje pavaizduotas medis yra trinaris.

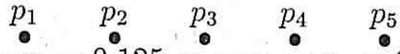
Tarkime, kad nusibraižėme kokį nors r-narį medį. Pagal šį medį sudarysime kodą iš $\mathcal{B} = \{b_1, b_2, \ldots, b_r\}$ abėcėlės žodžių. Jeigu nusibraižėme dvinarį (kaip 1 brėžinyje) medį, naudosime dvinarę abėcėlę $\mathcal{B} = \{0, 1\}$.

Pradėkime nuo šaknies. Iš jos išeinančioms šakoms priskirkime skirtingus simbolius. Po to tai padarykime šakoms išeinančioms iš kitų viršūnių, ir taip toliau. Galimybių yra ne viena.

Jeigu visoms šakoms jau priskyrėme po simbolį, grįžkime prie šaknies ir keliaukime nuo jos į kiekvieną lapą, rašydami paeiliui sutinkamus simbolius į eilutę. Kiekvieną kelią nuo šaknies iki lapo atitinka vienas abėcėlės $\mathcal{B}$ simbolių žodis. Pavyzdžiui, 1 brėžinio medį su pavaizduotu simbolių priskyrimu šakoms atitinka tokia žodžių aibė (kodas):

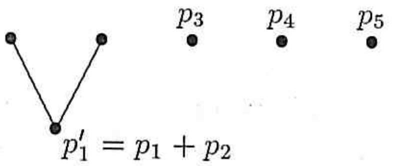
$$101;\ 111;\ 110;\ 01;\ 001.$$

Pastebėkime, kad šis kodų sudarymo būdas visada duoda tik p kodus, t. y. nė vienas žodis negali būti kito pradžia.

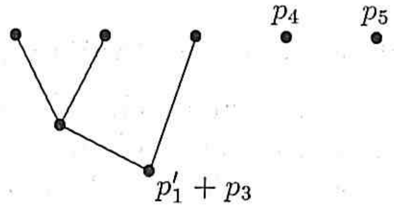


$(p_1 = p_2 = 0.125, p_3 = p_4 = p_5 = 0.25)$

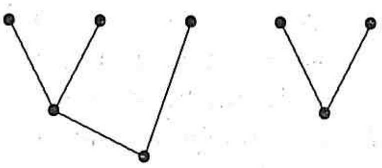
3.1 brėž.



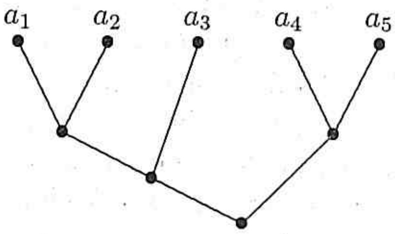
3.2 brėž.



3.3 brėž.



3.4 brėž.



3.5 brėž.

Dabar jau galime pereiti prie Huffmano kodų sudarymo. Juos konstruosime iš anksto nusibraižę kodo medį. Tarkime, šaltinis perduoda abėcėlės $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$ simbolius su tikimybėmis $p_1, p_2, \ldots, p_n$, o juos koduoti norime dvinarės abėcėlės žodžiais. Iš pradžių pažymėkime lapus, jų bus lygiai n, juos atitinka $\mathcal{A}$ simboliai, taip pat šių simbolių tikimybės (žr. 3.1 brėžinį). Suraskime mažiausias tikimybes (tegu tai būna p_1, p_2) ir išveskime iš jas atitinkančių viršūnių šakas į bendrą viršūnę, kaip parodyta 3.2 brėžinyje. Šiai bendrajai viršūnei priskirkime naują tikimybę $p_1' = p_1 + p_2$. Dabar ieškokime dviejų mažiausiųjų tikimybių tarp $p_1', p_3, \ldots, p_n$. Galbūt mažiausios bus p_1', p_3. Jas vėl sujunkime, naujajai viršūnei priskirkime tikimybę $p_1' + p_3$ ir taip toliau. Šitaip galų gale gausime dvinarį medį, kaip pavaizduota 3.5 brėžinyje. Toliau – viskas paprasta. Anksčiau aptartu būdu sudarykime kokį nors kodą.

Simbolis a_i koduojamas tuo žodžiu, kuris atitinka kelią, vedantį iš šaknies į simbolio a_i lapą.

O štai kas yra svarbiausia:

Huffmano algoritmu sudaryti kodai yra optimalūs!

Pasitreniruokite: kokiu dvinariu kodu geriausia koduoti simbolius, pasirodančius su tikimybėmis

$$0,05;\ 0,05;\ 0,1;\ 0,1;\ 0,2;\ 0,25;\ 0,25?$$

Nors Huffmano sprendimas nuostabus, gerų šaltinio kodų ieškojimo istorija tuo nesibaigia. Juk praktika visada kelia savų reikalavimų. Gerai, jei koduojamas šaltinis yra ištyrinėtas ir simbolių pasirodymo tikimybės žinomos. O jeigu ne? Jei koduoti tenka šaltinį, su kuriuo pirmąkart susiduriame? Taigi atsakymai visada kelia naujus klausimus.

Informacija ir triukšmas

Kas nesusierzino, neišgirdęs svarbios žinios dėl triukšmo? Triukšmą mes suprasime savotiškai: tai informacijos perdavimo kanalo savybė, dėl kurios informacija gali pasikeisti. Pavyzdžiui, į pažymių knygelę mokytoja įrašė 3, o pakeliui į namus 3 pavirto į 8. Nes kaltas čia triukšmas!

Kadangi kanalų be triukšmo nebūna, reikia sukti galvą, ką daryti, kad triukšmas neužgožtų svarbios informacijos. Mokytojai bepigu – ji gali pažymį koduoti raidėmis ir šitaip išvengti triukšmo įtakos. Elektroninių ryšių kanalais keliaujanti informacija – nulių ir vienetų srautai. Tačiau ir šiuo atveju mokytojos strategija efektyvi: prieš patikint vertingą informaciją perdavimo kanalui, reikia ją tinkamai paruošti (koduoti), atsižvelgiant į iškraipymo galimybes. Kaip tai daroma – svarbi teorija, o kartu ir praktiniai sprendimai. Panagrinėsime vos vieną paprastą situaciją. Kartais pakanka žinoti, kad įvyko klaida. Nustačius, kad priimta informacija klaidinga, kartais įmanoma paprašyti siuntėjo, kad pakartotinai perduotų tą pačią informaciją.

Visi daugybę kartų matėme ant prekių etikečių baltą stačiakampį, kuriame nubraižyta nevienodo storio juodų brūkšnių tvorelė. Po ja – 13 skaitmenų. Brūkšniais ir skaitmenimis užrašyta ta pati informacija apie prekę. Tai EAN (*European Article Numeration*) sistema. Skaičiai užrašyti žmogui, o brūkšniai – optiniam-elektroniniam skaitymo įrenginiui. Skaitant informaciją, pasitaiko klaidų. Kaip bent daugumos jų išvengti, kad klaidingai perskaityta informacija nesukeltų netvarkos prekių apskaitoje? Reikia, kad pats įrenginys galėtų signalizuoti, kad kažkas „įtartina“.

5 000443 008538 >

Panagrinėkime, kaip tai daroma EAN sistemoje. Čia informacija apie prekę užrašoma iš tiesų tik pirmaisiais 12 skaitmenų. Paskutinis – tryliktasis pridėtas tam, kad prietaisas (arba žmogus) galėtų pasitikrinti, ar informacija tikrai teisingai perskaityta. Skaitmenys $X_1, X_2, \ldots, X_{12}$, kuriais užrašyta prekės pagaminimo šalis bei kita informacija, ir kontrolinis skaitmuo X_{13}, susiję kontroline lygybe

$$X_1 + 3X_2 + X_3 + 3X_4 + \ldots + 3X_{10} + X_{11} + 3X_{12} + X_{13} \equiv 0(\text{mod } 10).$$

Žymuo $\equiv 0(\text{mod } 10)$ reiškia, kad lygybės kairiosios pusės skaičius turi dalytis iš 10. Taigi perskaitęs visus skaitmenis, įrenginys tikrina, ar teisinga kontrolinė lygybė. Jeigu ne – konstatuojama, kad informacija perskaityta klaidingai. Šis kontrolinio simbolio metodas visada nustato, kad įvyko klaida, jeigu neteisingai perskaitytas tik vienas simbolis. Jeigu neteisingai perskaityti du ar daugiau skaitmenų, klaidingas nuskaitymas gali būti nepastebėtas. Pavyzdžiui, jeigu skaitmuo X_1 perskaitomas kaip X_3, o X_3 – kaip X_1.

Vienas papildomas kodo simbolis leidžia kai kuriais atvejais nustatyti, kad perduodant įvyko klaidos. Gali kilti mintis, kad pridėjus daugiau simbolių, kodas taps dar „gudresnis“ ir sugebės daugiau. Teisinga mintis! Panašiai elektroninių skaičiavimo mašinų eros aušroje galvojo ir R. Hammingas: jei mašina sugeba surasti klaidą, kodėl negalėtų jos ir ištaisyti?

Paaiškinsime, kaip naudojantis Hammingo kodu galima ištaisyti įvykusią perdavimo klaidą.[4]

Sudarysime kodą 16 simbolių abėcėlei koduoti. Tuos simbolius užrašysime nulių ir vienetų ketvertais (žodžiais):

$$0000,\ 0001,\ 0010,\ \ldots,\ 1110,\ 1111.$$

Prieš perduodami žodį $x_1x_2x_3x_4$ į kanalą, jį pailginsime, pridėdami dar tris simbolius x_5, x_6, x_7, kurie sudaromi pagal tokias taisykles:

$$\begin{aligned} x_5 &= x_2 + x_3 + x_4, \\ x_6 &= x_1 + x_3 + x_4, \\ x_7 &= x_1 + x_2 + x_4; \end{aligned}$$

čia sudėtis atliekama moduliu 2, t. y. sudedant reikia naudotis taisyklėmis $0 + 0 = 1 + 1 = 0, 0 + 1 = 1$. Pavyzdžiui, jeigu norime perduoti žodį 1011, tai pridėję tris simbolius, į kanalą perduosime žodį

$$x_1x_2x_3x_4x_5x_6x_7 = 1011010. \qquad (5)$$

[4] Norintys suprasti, kodėl ši procedūra veikia, turėtų pavartyti kokią nors kodavimo teorijos knygą, pavyzdžiui, *V. Stakėnas, Informacijos kodavimas, Vilniaus universiteto leidykla, 1996.*

Į kanalą perduotas žodis $x_1x_2x_3x_4x_5x_6x_7$ gali būti iškraipytas, ir gavėją pasieks jau galbūt kitas žodis $x_1^*x_2^*x_3^*x_4^*x_5^*x_6^*x_7^*$. Tarkime, kanalas gali iškraipyti daugiausia vieną perduodamo žodžio $x_1x_2x_3x_4x_5x_6x_7$ simbolį. Pavyzdžiui, pasiuntus (5) žodį, gavėją pasiekė

$$x_1^*x_2^*x_3^*x_4^*x_5^*x_6^*x_7^* = 1010010. \tag{6}$$

Kaip nustatyti, kurie gauto žodžio simboliai teisingi, kurie ne? Mūsų kodas sukonstruotas taip, kad gavėjas pagal gautąjį žodį gali nustatyti, kurį simbolį reikia pakeisti. Naudodamas sudėtį moduliu 2, jis turi suskaičiuoti tris dydžius:

$$\begin{aligned} s_0 &= x_1^* + x_3^* + x_5^* + x_7^*, \\ s_1 &= x_2^* + x_3^* + x_6^* + x_7^*, \\ s_2 &= x_4^* + x_5^* + x_6^* + x_7^* \end{aligned}$$

ir sudaryti natūralųjį skaičių

$$k = s_0 \cdot 1 + s_1 \cdot 2^1 + s_2 \cdot 2^2.$$

Šis skaičius nurodo, kuris simbolis gautame žodyje $x_1^*x_2^*x_3^*x_4^*x_5^*x_6^*x_7^*$ yra klaidingas. Įsitikinkite (5), (6) žodžių pavyzdžiu, kad metodas tikrai „dirba“!