hyväksymispäivä arvosana

arvostelija

Lyhimmän kuvauspituuden periaate mallin valinnassa

Eric Andrews

Helsinki 6.5.2011

Kandidaatintutkielma

HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Laitos — Institution — Department
Matemaattis-luonnontieteellinen tiedekunta	Tietojenkäsittelytieteen laitos

Tekijä — Författare — Author
Eric Andrews

Työn nimi — Arbetets titel — Title

Lyhimmän kuvauspituuden periaate mallin valinnassa

Oppiaine — Läroämne — Subject
Tietojenkäsittelytiede

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Kandidaatintutkielma	6.5.2011	25 sivua

Tiivistelmä — Referat — Abstract

Lyhimmän kuvauspituuden periaate eli MDL-periaate on informaatioteoreettinen lähestymistapa induktiiviseen päättelyyn. Periaate perustuu suurilta osin algoritmisen informaatioteorian keskeisiin tuloksiin. MDL-periaatetta voidaan käytännössä soveltaa kilpailevien mallien vertailussa tai itse mallien valinnassa erilaisissa koneoppimis- ja hahmontunnistustehtävissä.

ACM Computing Classification System (CCS):
E.4 [Coding and information theory],
I.2.6 [Learning],
I.5.3 [Clustering],
A.1 [Introductory and Survey]

Avainsanat — Nyckelord — Keywords
MDL, Kolmogorov-kompleksisuus, induktiivinen päättely, mallin valinta, informaatioteoria

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — övriga uppgifter — Additional information

Sisältö

1 Johdanto

Lyhimmän kuvauspituuden periaate, lyhyemmin MDL-periaate (engl. *minimum description length principle)* on alkuaan suomalaisen Jorma Rissasen 1978 kehittämä periaate induktiiviseen päättelyyn [Ris78]. Induktiivisessa päättelyssä pyritään yleiseen johtopäätökseen rajallisesta havaintoaineistosta. Periaatetta sovelletaan useimmiten (tilastollisen) mallin valinnassa, mutta sitä voidaan soveltaa myös tilastollisessa ennustamisessa sekä tiedon tiivistämisessä [BRY98].

MDL-periaate perustuu havaintoon, jonka mukaan mitä tahansa havaintoaineistosta löydettävää säännöllisyyttä voidaan käyttää aineiston tiivistämiseen. Toisin sanoen havaintoaineisto voidaan ilmaista säännöllisyydet tuntien vähemmällä symbolimäärällä kuin mitä aineiston kirjaimelliseen kuvaukseen tarvittaisiin [Grü05]. MDL-periaatteen mukaan tulisi valita se säännöllisyyttä ilmaiseva malli, joka voidaan itse ilmaista vähällä vaivalla, mutta joka samalla mahdollistaa aineiston ilmaisemisen mahdollisimman tiivisti [HaY01].

Tieteellisessä tutkimuksessa havaitulle ilmiölle löydetään usein monta eri selitystä eli teoriaa. Kilpailevia teorioita vertaillaan, ja niistä pyritään valitsemaan paras. Valinnassa usein sovelletaan nk. Occamin partaveistä, jonka mukaan paras teorioista on se, joka on kaikista yksinkertaisin mutta myös riittävän selitysvoimainen [GLV00]. MDL-periaate kvantifioi Occamin partaveitsen ja tuo sen käytännön tilastolliseen päättelyyn huomioimalla, että yksinkertaisuudella on kaksi merkitystä: kuinka yksinkertaisesti teoria selittää ilmiön, ja kuinka yksinkertainen itse teoria on [KeL05].

MDL-periaatteen vahva informaatioteoreettinen pohja on tuonut uusia näkökulmia tilastolliseen päättelyyn. Rissasen MDL-periaatteessa kilpailevia selityksiä eli malleja ei niinkään nähdä aineiston taustalla olevan ”todellisen” mallin approksimaatioina, vaan malleja tutkitaan kuvailevasta näkökulmasta [MNP06]. Päämääränä on tunnistaa säännöllisyyksiä ja saada ilmiöstä uutta tietoa annettujen mallien pohjalta riippumatta siitä, ovatko ne uskottavia selityksiä ilmiölle vai ei [Grü05].

Tämä tutkielma koostuu MDL-periaatteen teoriasta, sekä periaatteen roolista erilaisissa käytännön sovelluksissa. Seuraavassa luvussa esitellään lyhyesti mistä mallin valinnassa on kyse. Sen jälkeen lukija johdatellaan MDL-periaatteeseen, joka on eräs ratkaisu mallin valinnan ongelmaan. Tutkielman loppupuolisko koostuu MDL-periaatteen sovelluksien esittelystä.

2 Mallin valinta

Aineistolla tarkoitetaan dataa, joka liittyy johonkin ilmiöön [MBB08]. Aineistona voi toimia esimerkiksi joukko mittaustuloksia, jotka esitetään koordinaatistossa, tai vaikkapa kokoelma luonnollisen kielen tekstejä. Myös kuvia, videoita ja ääntä voi käsitellä aineistona. Tärkeintä on, että aineisto voidaan järkevästi esittää merkkijonona tai numeerisessa muodossa.

Aineisto on yleensä vain pieni otos kaikesta ilmiöstä (teoriassa) mitattavissa olevasta datasta, jota nimitetään populaatioksi. Toivomuksena on, että aineisto on jossain mielessä kuvaava esimerkki populaatiosta. Tällöin aineistosta johdettavat säännöllisyydet ja mallit yleistyvät koskemaan myös koko populaatiota. Koko populaation läpikäyminen on harvoin mahdollista, varsinkaan kun tutkitaan todellisen maailman ilmiöitä, kuten esimerkiksi jonkin väestön kulutuskäyttäytymistä. Siispä joudutaan tyytymään otoksiin [MBB08].

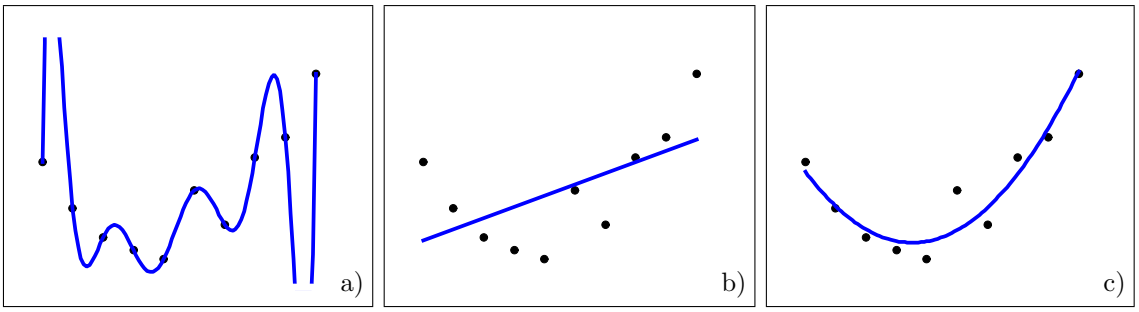
Tilastollinen malli on joukko todennäköisyysjakaumia, jotka ovat samaa funktionaalista muotoa. Jakaumat voidaan usein määritellä parametrivektorin θ avulla, joka on eräs alkio mallin parametriavaruudessa Θ. Tällöin tilastollinen malli voidaan ilmaista formaalisti seuraavasti: $\mathcal{M} = \{P_\theta \mid \theta \in \Theta\}$ [Grü07].

Todennäköisyyslaskennan peruskursseilta tuttu normaalijakauma on esimerkki tilastollisesta mallista, jonka yksittäisiä jakaumia saadaan kiinnittämällä parametriparin $\theta = (\mu, \sigma^2)$ arvot, missä μ on odotusarvo ja σ keskihajonta. Kellokäyrän sijaintia ja jyrkkyyttä voidaan siis säädellä parametreja muuttamalla.

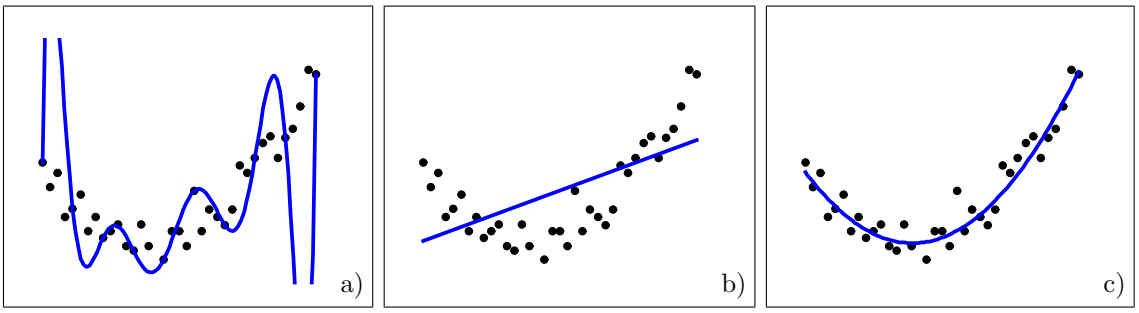
Tässä tutkielmassa, kun puhun yleisesti malleista, tarkoitan myös sellaisia, joihin ei suoranaisesti liity sattumaa. Esimerkiksi kaikki kolmannen asteen polynomit $a_3x^3 + a_2x^2 + a_1x + a_0$ muodostavat mallin, ja yksittäisiä polynomeja saadaan asettamalla kertoimet (a_0, a_1, a_2, a_3), missä $a_i \in \mathbb{R}$. Hypoteesiksi kutsutaan mallin ilmentymää, joka saadaan aikaiseksi kiinnittämällä mallin parametrien arvot. Joskus sanaa malli käytetään myös tarkoittamaan hypoteesia.

Tarve mallin valinnalle syntyy tilanteissa, joissa on joukko kilpailevia selityksiä eli malleja jollekin ilmiölle. Malli on valittava ilmiöstä mitatun aineiston pohjalta [HaY01]. Malleista pyritään valitsemaan se, joka Occamin partaveitsen mukaan parhaiten kuvaa aineiston takana piilevää ilmiötä.

Paras malli on tällöin sellainen, joka sovittuu hyvin annetun aineiston kanssa, mutta on kuitenkin tarpeeksi yksinkertainen ja yleinen selittääkseen ilmiötä ylipäätänsä. Siis keskenään tasapainotettavat suureet ovat mallin yksinkertaisuus (engl. *parsi-*



Kuva 1: Esimerkki a) polynomin ylisovituksesta, b) alisovituksesta ja c) hyvästä sovituksesta.



Kuva 2: Aineistoa vastaan sovitettujen polynomien sovittuminen koko populaatioon.

mony) ja hyvä sopivuus aineistoon (engl. *goodness-of-fit)* [For00].

Mallin valinnassa halutaan välttää kahta skenaariota, joita nimitetään ylisovittamiseksi (engl. *overfitting)* ja alisovittamiseksi (engl. *underfitting)*. Näiden kahden käsitteen havainnollistamiseksi oletetaan, että on joukko pisteitä, joihin tulee sovittaa jokin n-asteinen polynomi. Kyseessä on mallin valinta -tehtävä, jossa malleina on eri kertoimilla ja asteluvuilla varustettuja polynomeja, ja aineistona on koordinaatiston pisteet.

Ylisovittamisessa valitaan malli, joka on liian monimutkainen tutkittavaan ilmiöön nähden [Haw04]. Malli, joka ylisovittuu aineistoon kuvaa enemmänkin satunnaista kohinaa tai virhettä kuin itse taustalla olevaa suhdetta, ilmiötä tai säännöllisyyttä [Grü07].

Kuvan 1 kohdan a) polynomi on ylisovitettu aineistoon. Siinä esiintyvä 9-asteen polynomi sovittuu virheettömästi datan kymmeneen pisteeseen. Saatu polynomi ei kuitenkaan kerro mitään uutta ja merkittävää tutkittavasta ilmiöstä. Polynomi on jossain mielessä vain vaihtoehtoinen esitys aineiston pisteille.

Ylisovittuvan mallin suurin ongelma on siinä, että jos ilmiöstä mitataan uusi aineisto, ja tätä verrataan mallin antamiin arvoihin, virhe on todennäköisesti erittäin

suuri, jolloin malli ei kerro mitään merkittävää tutkittavasta populaatiosta [Grü07]. Ylisovitettu malli ei myöskään ennusta luotettavasti pisteiden välissä olevia tai viimeisen pisteen jälkeen tulevia pisteitä [Haw04].

Kuvan 2 kohdassa a) nähdään, kuinka aineistoon ylisovitettu polynomi sovittuu koko populaatioon. Kuvan vasemmalla ja oikealla puolella olevat suuret vaihtelut eivät selvästikään kuvaa populaation käyttäytymistä. Kuvan keskellä ilmenevä oskillaatio ei myöskään ole perusteltu populaation valossa.

Kuvan 1 kohdan b) polynomi taas alisovittuu pisteisiin. Suora on liian yksinkertainen malli, ja se ei kykene ilmaisemaan aineistossa esiintyvää säännöllisyyttä riittävän hyvin. Mallin antamat arvot poikkeavat suuresti aineiston pisteistä, ja samoin käy myös muilla ilmiön aineistoilla. Kuvan 2 kohdassa b) nähdään, että suora sovittuu populaatioon suunnilleen yhtä huonosti kuin aineistoon.

Tavoitteena on siis löytää jotain tältä väliltä, kuten kuvan 1 kohdan c) toisen asteen polynomi. Jos oletetaan, että valittu aineisto kuvaa hyvin populaatiota, niin virhe jää suhteellisen pieneksi muissakin aineistoissa, kuten näkyy kuvan 2 kohdassa c). Virhe on ainakin keskimäärin pienempi kuin se, joka saataisiin mallilla, joka yli- tai alisovittuu aineistoon.

3 Lyhimmän kuvauspituuden periaate

MDL-periaate perustuu keskeisesti algoritmisen informaatioteorian tuloksiin. Tässä luvussa esitellään ”idealisoitu” MDL-periaate, joka toimii käytännössä sovellettavan pohjana.

Tutkitaan seuraavaa kahta merkkijonoa:

$$ababababababababababababababababababababababababababab \quad (1)$$

$$baabaaababbaaababbaabbabbbabaaabbabbbbabaaaababababbabab \quad (2)$$

Molemmat merkkijonot koostuvat 54 merkistä, jotka ovat joko ’a’ tai ’b’. Ensisilmäyksellä ensimmäinen merkkijono on jossain intuitiivisessa mielessä yksinkertaisempi kuin toinen. Ensimmäinen merkkijono näyttäisi noudattavan jotain sääntöä, kun taas toinen vaikuttaa täysin satunnaiselta.

Ensimmäinen merkkijono voidaan kuvailla suomeksi vaikkapa seuraavasti: ”ab 27 kertaa”. Kuvaus koostuu 12 merkistä ja on lyhyempi kuin merkkijono, jota se kuvaa. Toisen merkkijonon ilmaiseminen tiiviimmin ei näytä onnistuvan millään ilmeisellä

tavalla, sillä merkkijonossa ei näytä olevan mitään säännöllisyyttä tai rakennetta, jota voitaisiin käyttää hyväksi kuvailussa. Tästä syystä toisen merkkijonon lyhin kuvaus on sen kirjaimellinen esitys.

MDL-periaate perustuu keskeisesti edellä esitettyyn havaintoon. Jos annetusta aineistosta löytyy säännöllisyyttä, niin aineistoa voidaan tiivistää tämän avulla. Aineisto voidaan siis kuvata vaihtoehtoisella tavalla niin, että käytetään vähemmän merkkejä kuin mitä käytettäisiin aineiston kirjaimelliseen kuvaukseen. Mitä enemmän aineistoa pystytään *tiivistämään*, sitä enemmän siitä ollaan *opittu*, eli sitä enemmän siitä on löydetty säännöllisyyttä [Grü05].

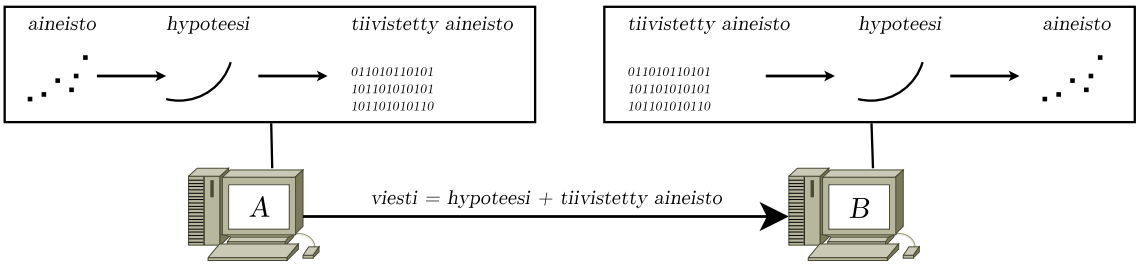
Ensimmäinen merkkijono voitaisiin ilmaista myös ”ab kaksikymmentäseitsemän kertaa”, ”ab 27 times” tai ”for(i=0;i<27;++i) print ’ab’; halt;”. Jokaisen kuvauksen pituus on kuitenkin eri. Jotta merkkijonojen kuvaamisesta ja kuvauksien vertailusta voitaisiin puhua mielekkäästi, tulee ensin määrittää ja kiinnittää jokin kuvausmenetelmä. Halutaan kuvausmenetelmä, jossa kuvauksista voidaan yksiselitteisesti päätellä alkuperäinen kuvattava kohde [Grü07].

Ray Solomonoff [Sol64] ehdotti uraauurtavassa artikkelissaan vuonna 1964, että kuvausmenetelmänä käytettäisiin universaaleja tietokonekieliä. Tällöin jokainen merkkijono D voitaisiin esittää tietokoneohjelmana P, joka tulostaa merkkijonon D ja pysähtyy. Solomonoffin tutkimuksen pohjalta Kolmogorov ja Chaitin kehittivät Kolmogorov-kompleksisuuden [WaD99].

Kolmogorov-kompleksisuus [Kol65, Cha66] on jonkin merkkijonon monimutkaisuuden mitta. Merkkijonon Kolmogorov-kompleksisuus on yhtä kuin sen lyhimmän kuvauksen pituus jossain kiinnitetyssä universaalissa kuvauskielessä. Universaali kuvauskieli voi olla esimerkiksi universaali Turingin kone tai jokin riittävän ilmaisukykyinen ohjelmointikieli kuten Python, C++ tai Java [WaD99].

Universaalin Turingin koneen yhteydessä Kolmogorov-kompleksisuuden voi muotoilla myös seuraavasti. Olkoon S kuvattava kohde ja I eräs sen kuvaus. Mikä on lyhin sellainen merkkijonosyöte I, että kun se annetaan syötteeksi universaalille Turingin koneelle T, niin T tulostaa täsmälleen S ja pysähtyy [WaD99]?

Kolmogorov-kompleksisuus on ajatukseltaan varsin selkeä, mutta sen soveltaminen käytännön induktiiviseen päättelyyn on ongelmallista, sillä Kolmogorov-kompleksisuus on ratkeamaton. Ei ole olemassa algoritmia, joka kertoisi annetun merkkijonon lyhintä kuvausta tai edes lyhimmän kuvauksen pituutta. Kolmogorov-kompleksisuutta ei voida siis sellaisenaan hyödyntää induktiivisessa päättelyssä [LiV08].



Kuva 3: Vastaanottajan on kyettävä purkaamaan viesti.

MDL-periaate tuo Kolmogorov-kompleksisuuden käytäntöön ottamalla käyttöön kuvausmenetelmiä, jotka ovat ilmaisukyvyltään rajoittuneempia kuin universaali Turingin kone. Kuvausmenetelmää on rajoitettava niin, että mille tahansa annetulle aineistolle eli merkkijonolle voidaan laskea lyhin kuvaus kuvausmenetelmässä [Grü07].

Rajoitetun kuvausmenetelmän huonona puolena on se, että tulee aina löytymään säännöllisiä merkkijonoja, joita menetelmässä ei pystytä tiivistämään. Toisaalta tiedetään, että tiiviimmän kuvauksen laskeminen mielivaltaiselle merkkijonolle on ratkeamaton ongelma. Käytännössä MDL-periaatetta soveltaessa voidaan usein valita sopiva kuvausmenetelmä ongelman mukaan [Grü07]. Esimerkiksi edellä esitellyssä polynomin sovitustehtävässä voitaisiin valita polynomimalleihin perustuva kuvausmenetelmä.

MDL-periaatteessa kuvausmenetelminä käytetään tilastollisia malleja, eli Turingin kone korvataan äärellisellä joukolla tilastollisia malleja [HaY01]. Kuvausmerkkijonot tulkitaan kaksiosaisiksi. Ensin määritellään hypoteesi, eli mitä mallia käytetään ja millä parametreilla. Sitten ilmoitetaan hypoteesin avulla tiivistetty aineisto. Näin tuodaan Kolmogorov-kompleksisuuden ja muun informaatioteorian tulokset mallin valinta -tehtäviin [WaD99].

Kuvausmenetelmän ehdoksi asetettiin, että alkuperäiset kohteet voidaan purkaa niiden (kuvausmenetelmällä saaduista) kuvauksista. Tämä toteutuu kaksiosaisessa kuvauksessa. Asiaa voi havainnollistaa ajattelemalla viestiprotokollaa, joka lähettää viestin koneelta A koneelle B kuten kuvassa 3. Ensin koneessa A laaditaan viesti eli kuvaus MDL-periaatteen mukaisesti tiivistämällä aineisto hypoteesin avulla, ja lähettämällä sekä hypoteesi että tiivistetty aineisto viestinä koneelle B. Kun B vastaanottaa viestin, siitä ensin puretaan hypoteesi (malli ja parametrit), ja tämän avulla puretaan tiivistetty aineisto takaisin alkuperäiseen muotoonsa [KMU95].

3.1 Määritelmä ja ominaisuudet

Kuten edellisessä luvussa tuli ilmi, MDL-periaatteessa käytetään kaksiosaisia kuvauksia aineiston tiivistämiseen hypoteesin avulla. Toivomuksena on tietenkin, että tämä kaksiosainen kuvaus on lyhyempi kuin aineiston kirjaimellinen esitys. Jotta näin voi tapahtua, on hypoteesin kyettävä selittämään aineistossa ilmenevää säännöllisyyttä lyhyemmässä muodossa kuin se esiintyy aineistossa.

Vaihtoehtoisia hypoteeseja tietylle aineistolle löytyy monta, joista jokainen selittää aineiston paremmin tai huonommin kuin muut. MDL-periaatteen näkökulmasta paras hypoteeseista on se, jonka kaksiosainen esitys on lyhin.

Määritelmä 1: kaksiosainen MDL-periaate hypoteesin valintaan [Grü07]
Olkoon $\mathcal{M}$ joukko hypoteeseja (esim. todennäköisyysjakaumia) ja D annettu aineisto. MDL-periaatteen mukaan tulee valita hypoteeseista $H \in \mathcal{M}$ se, joka minimoi summan

$$L(H) + L(D|H).$$

Tässä termi $L(H)$ on hypoteesin esittämiseen tarvittavien merkkien määrä, ja $L(D|H)$ on hypoteesin avulla tiivistetyn aineiston esittämiseen tarvittavien merkkien määrä. Merkistönä toimii yleensä binääriaakkosto $\{0, 1\}$.

Edellinen määritelmä koski hypoteesin valintaa, jossa pyritään valitsemaan mallin lisäksi optimaalisimmat parametriarvot. Jos tavoitteena on ainoastaan vertailla eri malleja keskenään, on tämäkin toki mahdollista määritelmän 2 tapaan.

Määritelmä 2: kaksiosainen MDL-periaate mallin valintaan
Koostukoon joukko $\mathcal{M} = \mathcal{M}_1 \cup \mathcal{M}_2 \cup ... \cup \mathcal{M}_n$ useamman mallin hypoteesien yhdisteestä (esim. 1-asteen, 2-asteen ja n-asteen polynomit). Tällöin MDL-periaatteen mukaan paras malli on se, johon MDL-periaatteella valittu hypoteesi H kuuluu.

Heti määritelmästä 1 nähdään, että MDL-periaatteessa joudutaan tasapainottamaan hypoteesin esityksen ja tiivistetyn aineiston pituuksien välillä. Näin vältytään valitsemasta hypoteesia, joka yli- tai alisovittuu annettuun aineistoon [Grü05] Yksinkertainen hypoteesi voidaan esittää vähällä merkkimäärällä, mutta usein aineiston tiivistyksen pituus jää suureksi, koska hypoteesi ei kykene käyttämään hyväksi riittävästi aineistossa ilmenevää säännöllisyyttä. Monimutkainen hypoteesi kykenee

sovittumaan paremmin aineistoon, ja näin vangitsemaan siinä ilmenevää säännöllisyyttä. Aineiston esitys saadaan erittäin tiiviiksi, mutta samalla hypoteesin esittämiseen tarvittavien merkkien lukumäärä kasvaa.

Palataan jälleen luvun 2 polynomin sovitus -tehtävään. Polynomit eli hypoteesit ilmaisevat aineiston säännöllisyyttä antamalla polynomin mukaiset arviot aineiston pisteistä. Aineisto tiivistetään ilmoittamalla todellisten pisteiden etäisyys polynomin antamiin vastaaviin arvoihin. Ylisovittuvan polynomin tapauksessa termi $L(D|H)$ jää pieneksi, koska polynomin antamat arvot osuvat aina virheettömästi pisteisiin. Toisaalta 9-asteen polynomin määrittelemiseksi on määriteltävä 10 eri termin arvo, jolloin $L(H)$ on suuri.

Alisovittuvan suoran tapauksessa $L(H)$ jää pieneksi, sillä on vain kaksi määriteltävää termiä. Toisaalta suuren poikkeaman vuoksi aineiston tiivistäminen ei tuota lyhyttä $L(D|H)$. Hyvin sovittuva toisen asteen polynomi saa kohtalaiset arvot molempiin termeihin. MDL-periaatteen nojalla valitaan hyvin sovittuva polynomi, jos sen ja aineistoon ilmaisemiseen tarvittavien merkkien määrä on pienempi kuin yli -ja alisovittuvan polynomin tapauksessa.

3.2 Aineiston tiivistäminen hypoteesin avulla

Aineiston tiivistäminen toteutetaan laatimalla sille koodi, joka perustuu käytettävään hypoteesiin. Koodin avulla aineisto voidaan esittää tiiviimmin, ja hypoteesin avulla koodattu aineisto voidaan purkaa takaisin alkuperäiseksi. Koodin laatiminen onnistuu helposti kun hypoteesit ovat todennäköisyysjakaumia. Koodausteoriassa (engl. *coding theory)* Kraftin epäyhtälön [Kra49] seurauksena saadaan bijektio todennäköisyysjakaumien ja koodien välille.

Formaalisti (epäsingulaarinen) koodi (engl. *(nonsingular) code)* on injektiivinen kuvaus $C : \mathcal{X} \to \{0,1\}^*$ lähdeaakkostosta $\mathcal{X}$ binääriseen koodisanajoukkoon. Injektiivisyys tarkoittaa, että $C(x) = z$ pätee korkeintaan yhdelle $x \in \mathcal{X}$. Seurauksena tästä jokainen koodisana voidaan purkaa yksikäsitteisesti takaisin aakkoston $\mathcal{X}$ alkioksi eli $C^{-1}(z) = x$. [CoT06].

Jokaisella koodilla C on koodipituus (engl. *codelength function)*, joka on kuvaus aakkostosta $\mathcal{X}$ luonnollisiin lukuihin eli $L_C : \mathcal{X} \to \mathbb{N}$. $L_C(x)$ kertoo, kuinka monta bittiä tarvitaan alkion x koodaukseen koodissa C; tarkemmin, kun x kuvataan koodilla C, niin kuinka monta bittiä maalijoukon vastaavassa binäärijonossa eli koodisanassa on [CoT06]. Siis formaalisti merkiten pätee $L_C(x) = |C(x)|$.

Otetaan nyt esimerkiksi aakkosto $\mathcal{X} = \{a, b, c, d\}$ jolle halutaan laatia koodi. Eräs koodi voisi olla esimerkiksi seuraava: $C_1(a) = 0$, $C_1(b) = 1$, $C_1(c) = 01$ ja $C_1(d) = 11$. Nyt voisimme päätellä esimerkiksi, että $C_1^{-1}(01) = c$ ja $L_{C_1}(a) = 1$.

MDL-periaatteen tapauksessa tavoitteena on laatia koodi, jolla voidaan esittää aineisto. Aineisto on merkkijono $D = a_1a_2...a_n$ aakkoston $\mathcal{X}$ symboleita [Grü00]. Merkkijonon koodaaminen tapahtuu koodaamalla jokainen merkki erikseen, ja katenoimalla saadut koodisanat yhteen: $C(a_1a_2...a_n) = C(a_1)C(a_2)...C(a_n)$ [CoT06].

Edellä esitetyssä koodissa C_1, koodisanojen katenaatioita ei voida aina purkaa yksikäsitteisesti takaisin aineistoksi. Esim. koodisanan 011 purkaminen voi tuottaa yhtä hyvin merkkijonot *abb*, *ad* tai *cb*. MDL-periaatteen kannalta on kuitenkin tärkeää, että kuvausmenetelmässä voidaan purkaa tiivistetty aineisto yksikäsitteisesti takaisin alkuperäiseen muotoonsa.

Yksikäsitteisyyden saavuttamiseksi MDL-periaatteessa huomioidaan vain prefiksikoodeja (engl. *prefix codes)*, jotka ovat kaikkien mahdollisten koodien eräs osajoukko [Grü05]. Prefiksikoodeissa mikään koodisana ei esiinny toisen etuliitteenä. Olkoon $B \subset \{0, 1\}^*$ jonkin prefiksikoodin maalijoukon koodisanat. Tällöin kaikille $b \in B$ pätee, että ei ole olemassa koodisanaa $g \in B$ niin että $g = bc$ jollain $c \in \{0, 1\}^+$ [CoT06].

Jatketaan aiempaa esimerkkiä laatimalla aakkostolle $\mathcal{X} = \{a, b, c, d\}$ prefiksikoodi C_2. Asetetaan koodisanat seuraavasti: $C_2(a) = 00$, $C_2(b) = 01$, $C_2(c) = 10$ ja $C_2(d) = 11$. Nyt voidaan purkaa koodisanajono 101100 yksikäsitteisesti, ja saadaan siis *cda*.

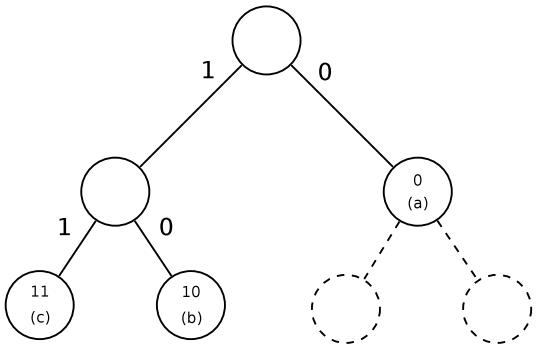
Aineistolle voidaan helposti laatia prefiksikoodi hypoteesin avulla, jos hypoteesi on todennäköisyysjakauma. Tämä tulos saadaan seurauksena seuraavaksi esitettävästä Kraftin epäyhtälöstä.

Kraftin epäyhtälö (Kraft-McMillan-teoreema) [CoT06] Olkoon $\mathcal{X}$ äärellinen aakkosto ja C prefiksikoodi. Tällöin (a) aakkoston symbolien koodipituudet noudattavat epäyhtälöä

$$\sum_{x \in \mathcal{X}} 2^{-L_C(x)} \leq 1, \tag{3}$$

ja kääntäen (b) koodipituuksia sisältävälle joukolle, joka toteuttaa epäyhtälön 3, on olemassa prefiksikoodi C, jonka koodisanoilla on koodipituusjoukon mukaiset pituudet.

Todistuksen hahmotelma [Grü07, CoT06] Perustellaan ensin suunta (a). Koodi voidaan esittää binääripuuna, jossa jokaisella solmulla on 0, 1 tai 2 lasta. Kaari



Kuva 4: Binääripuuesitys aakkoston {a,b,c} prefiksikoodille C, jossa $C(a) = 0$, $C(b) = 10$ ja $C(c) = 11$ [Grü07].

vasempaan lapseen vastaa ykköstä ja kaari oikeaan nollaa. Jokainen koodisana vastaa sitä solmua puussa, johon päästään seuraamalla koodisanan määrämää polkua. Esimerkiksi kuvassa 4 koodisanaa 10 vastaa solmu, johon päästään kun mennään ensin juuren vasempaan lapseen, ja sitten saavutun solmun oikeaan lapseen.

Selvästi koodipituudet $l_x = L_C(x)$ vastaavat polkujen pituuksia puussa. Merkitään pisintä koodisanaa ja polkua l_{max}, ja laajennetaan puuta niin, että jokainen eri l_{max}-pituinen polku on mahdollinen.

Koska koodina on prefiksikoodi niin tiedetään, ettei koodisanallisella solmulla voi olla jälkeläisiä, joihin liittyisi koodisana. Jokaisella aakkoston $x \in \mathcal{X}$ vastaavalla koodisanalla on alipuu, joka koostuu itse koodisanaa vastaavasta solmusta, ja tämän jälkeläisistä. Jos esimerkiksi $l_x = l_{max}$, niin alipuu koostuu vain koodisanaa vastaavasta solmusta, jos taas $l_x = l_{max} - 1$, niin alipuu koostuu koodisanasolmun lisäksi tämän välittömistä lapsista.

Koostukoon joukko D_x symbolin $x \in \mathcal{X}$ koodisanan alipuun lehtisolmuista. Nyt pätee $2^{l_x} \cdot |D_x| = 2^{l_{max}}$, eli että $2^{-l_x} = |D_x| \cdot 2^{-l_{max}}$. Lisäksi koska prefiksikoodista seuraa, että $D_x \cap D_y = \emptyset$ kaikilla $x, y \in \mathcal{X}$, missä $x \neq y$, voidaan päätellä, että

$$\sum_{x \in \mathcal{X}} 2^{-l_x} \;=\; \sum_{x \in \mathcal{X}} |D_x| \cdot 2^{-l_{max}} \;=\; 2^{-l_{max}} \sum_{x \in \mathcal{X}} |D_x| \;\leq\; 2^{-l_{max}} \cdot 2^{l_{max}} = 1.$$

Perustellaan vielä käänteinen suunta (b). Olkoon annettu koodipituudet $l_1, l_2, ..., l_m$, jotka toteuttavat Kraftin epäyhtälön. Tällöin voidaan rakentaa koodipituuksia vastaava binääripuu kuvan 4 tapaan, jossa jokaista koodipituutta vastaa jokin solmu eli koodisana. Valitaan syvyyden l_1 leksikografisesti ensimmäinen solmu 1 vastaavaksi koodisanaksi. Valitaan syvyyden l_2 leksikografisesti ensimmäinen solmu, joka

ei ole 1 vastaavan solmun jälkeläinen, 2 vastaavaksi koodisanaksi. Näin jatkaen voidaan rakentaa prefiksikoodi, jossa jokaista $i \in \{1, ..., m\}$ vastaa jokin l_i pituinen koodisana.

□

Seuraus [Grü00] Kaikille äärellisen aakkoston $\mathcal{X}$ prefiksikoodeille C on olemassa todennäköisyysjakauma P_C siten, että kaikille symboleille $x \in \mathcal{X}$ pätee

$$P_C(x) = 2^{-L_C(x)}.$$

Kaikille todennäköisyysjakaumille P, jonka arvojoukko $\mathcal{X}$ on äärellinen, on olemassa prefiksikoodi C_P siten, että kaikille $x \in \mathcal{X}$ pätee

$$L_{C_P}(x) = \lceil -\log_2 P(x) \rceil.$$

Seuraus antaa tavan määritellä todennäköisyysjakauma, kun tunnetaan ainoastaan prefiksikoodi, ja toisaalta tavan määritellä prefiksikoodipituudet, kun tunnetaan vain jakauma. Selvästi suuret todennäköisyydet vastaavat lyhyitä koodipituuksia, ja pienet todennäköisyydet suuria koodipituuksia.

Olkoon esimerkiksi aakkostolle $\mathcal{X} = \{a, b, c, d\}$ määritelty prefiksikoodi C sellainen, että $a \mapsto 0$, $b \mapsto 11$, $c \mapsto 100$ ja $d \mapsto 101$. Tällöin saadaan laskettua vastaava todennäköisyysjakauma P_C, jossa $P_C(a) = 2^{-L_C(a)} = \frac{1}{2}$, $P_C(b) = \frac{1}{4}$ ja $P_C(d) = P_C(c) = \frac{1}{8}$.

Seurauksen avulla voidaan myös laskea todennäköisyysjakaumalle koodipituudet, ja Shannon-Fano-koodauksella voidaan määrittää prefiksikoodi. MDL-periaatteessa kuitenkin jo koodipituudet riittävät, sillä käytännössä mitään koodaamista ei tarvitse tehdä. Tästä syystä koodipituuksia ei välttämättä tarvitse pyöristää ylös kokonaisluvuiksi, vaan voidaan käyttää suoraan idealisoituja koodipituuksia $L_{C_P}(x) = -\log_2 P(x)$ [MNP06].

Prefiksikoodia vastaavan todennäköisyysjakauman todennäköisyysmassa ei aina summaudu tasan yhdeksi, eikä se tästä syystä tarkalleen ottaen aina ole todennäköisyysjakauma. Puuttuva massa voidaan kuitenkin jakaa kuvitteelliselle, ylimääräiselle ja aakkostoon kuulumattomalle symbolille $\diamond \notin \mathcal{X}$ siten, että $\sum_{x \in \mathcal{X} \cup \{\diamond\}} P_C(x) = 1$ [Grü05].

Todennäköisyysjakaumia ajatellaan tässä vain ja ainoastaan matemaattisina objekteina, eikä oleteta, että aineiston ja koodin (tai jakauman) välillä olisi välttämättä

stokastista suhdetta. Erityisesti ei oleteta, että aineiston koodaamiseen käytettävä koodi vastaisi aineiston ”todellista” dataa generoivaa jakaumaa [Grü00].

Tässä luvussa esitelty vastaavuus koodipituuksien ja todennäköisyysjakaumien välillä on tärkeä osa MDL-teoriaa. Usein hypoteesit ovat todennäköisyysjakaumia, tai ne voidaan helposti muuntaa sellaisiksi. Tiivistetyn aineiston pituuden $L(D|H)$ laskeminen hypoteesilla, joka vastaa todennäköisyysjakaumaa onnistuu helposti laskemalla aineiston todennäköisyys jakaumassa, ja muuntamalla saatu arvo koodipituudeksi. Luvun 4.2 sovelluksessa tätä yhteyttä käytetään monessa kohtaa hyväksi.

3.3 Hypoteesin esittäminen

Kaksiosaisen MDL-periaatteen heikkous on hypoteesin esityksen pituuden määrittäminen. Esityksen pituus voi vaihdella suuresti eri hypoteesin koodaustapojen välillä, ja vaarana on, että hypoteesin koodauksesta tulee täysin mielivaltaista. Valitettavasti ei ole mitään eheää matemaattista pohjaa, jonka mukaan $L(H)$ voitaisiin laskea siististi. Käytännössä tyydytäänkin erilaisiin ad hoc -laskutapoihin sen mukaan, minkälaista hypoteesia ollaan esittämässä [Grü05].

Käytännön sovelluksissa pyritään laatimaan esityksen pituudet niin, että monimutkaisemmat hypoteesit ovat pitempiä kuin yksinkertaisemmat. Esimerkiksi polynomien tapauksessa voidaan koodata hypoteesi ilmoittamalla asteluku ja termien kertoimet jollain rajatulla tarkkuudella. Monimutkaisempi eli korkeamman asteen polynomi vaatii useamman kertoimen koodaamisen kuin yksinkertaisen polynomin.

Jorma Rissanen [Ris84] huomasi vuonna 1984, että hypoteesin esityksen ongelman voi väistää käyttämällä kaksiosaisten esitysten sijaan yksiosaisia, jotka perustuvat universaaleihin koodeihin. Modernissa MDL-periaatteessa hypoteesin esityksen pituuteen liittyvää epävarmuutta ei ole. Tässä tutkielmassa ei kuitenkaan syvennytä moderniin MDL-periaatteeseen, sillä sen ymmärtäminen vaatii suhteellisen laajaa informaatioteoreettista pohjaa.

3.4 Yksinkertainen esimerkkisovellus

Seuraavaksi katsotaan, kuinka MDL-periaatetta voidaan soveltaa yksinkertaiseen kolikon heitto -ongelmaan. Tarkoituksena on konkretisoida MDL-periaatteen toimintaa ennen kuin esitellään varsinaisia todellisen maailman sovelluksia. Tässä esitettävä esimerkki on muunnelma Hansenin ja Yunin katsauksessa [HaY01] esitetystä

esimerkistä.

Kolmea kolikkoa heitettiin 60 kertaa, ja saadut tulokset kirjattiin ylös merkkijonoina, joissa 1 vastaa kruunaa ja 0 klaavaa:

$$d_1 = 111011110111111001111110111111110100111111101111111111111011$$
$$d_2 = 010001100000111010100011101001100101000110001101010110110100$$
$$d_3 = 101010101010101010101010101010101010101010101010101010101010$$

Tehtävänä on selvittää, olivatko heitetyt kolikot painotettuja vai ei. Kolikon heiton mallina toimii Bernoullin jakaumat $X \sim B(\theta)$, missä $P(X = 1) = \theta$ vastaa kruunan ja $P(X = 0) = 1 - \theta$ klaavan todennäköisyyttä.

MDL-periaatetta sovelletaan valitsemalla se hypoteesi, eli parametri $\theta \in \Theta$, joka minimoi kaksiosaisen esityksen $L(H) + L(D|H)$. Hypoteesi voidaan esittää tässä tapauksessa pelkästään ilmoittamalla parametrin θ arvo, sillä käytössä on vain yksi tilastollinen malli. Rajoitutaan parametreihin, jotka ovat muotoa $\theta = \frac{m}{60}$, missä $m = \{0, .., 60\}$. Nyt hypoteesin esityksen pituudeksi saadaan $L(H) = \log_2 61 \approx 6$, sillä muuttujan m arvo voidaan valita 61 eri vaihtoehdosta.

Yhden merkkijonon $d_i \sim X_1, X_2, ..., X_{60}$ merkit (kolikon heitot) ovat toisistaan riippumattomia ja samoin jakautuneita, joten jonon todennäköisyys voidaan laskea seuraavasti: $P(\underline{X} = d_i) = \theta^k(1-\theta)^{60-k}$, missä k on jonon d_i kruunien lukumäärä. Aiemmin esitellyn Kraftin epäyhtälön nojalla aineiston koodipituudeksi saadaan:

$$\begin{aligned} L(D|H) = L(d_i|\theta) &= -\log_2 P_\theta(\underline{X} = d_i) = -\log_2(\theta^k(1-\theta)^{60-k}) \\ &= -\log_2 \theta^k - \log_2(1-\theta)^{60-k} = -k\log_2(\theta) - (60-k)\log_2(1-\theta) \end{aligned}$$

Oletetaan, että kolikot voivat olla painotettuja vain kolmella eri tavalla, ja asetetaan parametriavaruudeksi $\Theta = (\frac{11}{60}, \frac{30}{60}, \frac{49}{60})$. Lasketaan ensimmäisen merkkijonon kaksiosainen koodipituus näillä kolmella vaihtoehdolla:

$$\begin{aligned} L(\frac{11}{60}) + L(d_1|\frac{11}{60}) &= 6 - 50\log_2(\frac{11}{60}) - 10\log_2(\frac{49}{60}) \approx 131 \\ L(\frac{30}{60}) + L(d_1|\frac{30}{60}) &= 6 - 50\log_2(\frac{1}{2}) - 10\log_2(\frac{1}{2}) = 66 \\ L(\frac{48}{60}) + L(d_1|\frac{49}{60}) &= 6 - 50\log_2(\frac{49}{60}) - 10\log_2(\frac{11}{60}) \approx 45 \end{aligned}$$

MDL-periaatteen mukaan tämän sarjan kolikko oli luultavasti painotettu kruunalle suhteessa 49 : 11. Merkkijonoille d_2 ja d_3 voidaan laskea koodipituudet samaan

tapaan, ja molemmat ovat MDL-periaatteen nojalla tasapainoisen kolikon heiton sarjoja eli $\theta = \frac{30}{60}$.

Merkkijono d_3 voitaisiin ilmaista vielä lyhyemmin mallilla, joka kykenee ilmaisemaan paremmin siinä ilmenevää säännöllisyyttä, kuten esimerkiksi Markovin ketjulla. Bernoullin jakauma ei kykene ilmaisemaan ajatusta ”jono ’10’ 30 kertaa”, jonka takia se ei itse asiassa edes kykene tiivistämään aineistoa. Itse asiassa jonon d_3 kirjaimellinen esitys vaatii vain 60 bittiä, kun taas edellä esitetyllä tavalla koodattuna sen esittäminen vaatii noin 66 bittiä.

Jos parametriavaruus Θ avataan kaikille $m \in \{0, ..., 60\}$, niin d_2 saavuttaa lyhimmän kaksiosaisen esityksensä kun $\theta = \frac{27}{60}$, koska jonossa d_2 on 27 ykköstä. Tällöin $L(\frac{27}{60}) + L(d_2|\frac{27}{60}) \approx 65$, joka on kuitenkin suurempi kuin aineiston kirjaimellinen esitys. Tämän voi tulkita niin, ettei aineistossa olevien kruunien määrä ole vielä riittävän suuri tukemaan väitettä, jonka mukaan kolikko olisi painotettu. Jonossa d_1 sen sijan esiintyy suhteessa niin paljon enemmän kruunaa kuin klaava, että väittämä painotetusta kolikosta on jo riittävän uskottava.

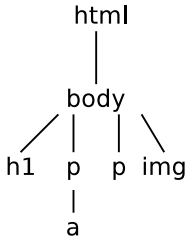
4 Sovellukset

MDL-periaate on kiinnostanut tutkijoita niin tietojenkäsittelytieteessä kuin tilastotieteessä, sekä tilastotiedettä soveltavilla aloilla, kuten ekonometriassa, biologiassa, psykologiassa ja sosiologiassa. Tietojenkäsittelytieteessä MDL-periaatetta sovelletaan erilaisiin koneoppimis- ja hahmontunnistustehtäviin.

4.1 Ryvästäminen

MDL-periaatetta on sovellettu paljon erilaisiin ryvästysongelmiin. Ryvästämisessä pyritään jakamaan joukko havaintoja osajoukkoihin, eli ryväisiin (engl. *cluster)* niin, että jokaisen ryvään havainnot ovat keskenään jollain mitalla samanlaisia.

Böhm ja kumppanit esittelevät [BFP06] MDL-periaatteeseen perustuvan RIC-kehyksen (engl. *robust information-theoretic clustering)* ryvästykseen. Kehyksen avulla toisella menetelmällä saatua ryvästystä voidaan parantaa poistamalla ylimääräinen kohina. Menetelmän etuna on, ettei se vaadi yhdenkään ylimääräisen parametrin määrittämistä ja se skaalautuu hyvin datan kasvaessa. Menetelmä vaikuttaa lupaavalta artikkelissa esitettyjen empiiristen kokeiden valossa.



Kuva 5: Yksinkertaisen HTML-dokumentin DOM-puu.

MDL-periaatetta voidaan käyttää apuna myös hieman käytännönläheisemmissä ryvästystehtävissä. Internet-sivustot koostuvat usein sivuista, jotka ovat HTML-rakenteeltaan identtisiä tai hyvin samanlaisia. Esimerkiksi useiden uutisivustojen uutiset ovat rakenteeltaan hyvin samanlaisia, vain sisältö (otsikko, kuva, leipäteksti) on eri. On esitetty monia tekniikoita, joita käyttäen dataa voidaan louhia tällaisilta sivuilta ja esittää vaikka XML-muodossa. Valitettavasti sivujen valitseminen ja syöttäminen tällaisiin järjestelmiin tehdään usein käsin.

Crescenzi ja kumppanit [CMM05] pyrkivät automatisoimaan valitsemisen algoritmilla, jolle annetaan syötteeksi lähtösivu, jonka jälkeen algoritmi käy läpi sivustoa ja pyrkii identifioimaan sekä sijoittamaan rakenteisesti samanlaisia sivuja samoihin ryväisiin. Saatuja ryväitä voidaan edelleen ohjata louhittavaksi.

Crescenzin ja kumppaneiden esittämä menetelmä perustuu havaintoon, että sivulla olevat linkit vastaavat sivun rakenteen säännöllisyyttä. Linkkejä usein tarjotaan linkkikokoelmana, ja kokoelman linkit ohjaavat usein rakenteisesti samanlaisille sivuille. Toisaalta samanlaisista linkkikokoelmista koostuvat sivut ovat usein myös muulta rakenteeltaan samanlaisia.

HTML-dokumenttia voidaan sen hierarkkisen rakenteen vuoksi käsitellä puuna kuten kuvassa 5. Tällaista puuta kutsutaan DOM-puuksi (engl. *document object model tree)*. DOM-puun polkua kutsutaan DOM-poluksi. Crescenzin ja kumppaneiden menetelmässä jokaiselle sivulle laaditaan skeema, jossa käy ilmi DOM-polut kaikkiin sivun linkkikokoelmiin. Esimerkki (yksinkertaistetun) sivun skeemasta on: {html-body-table-tr-td, html-body-div-ul-li, html-body-div-p}.

Algoritmin toiminta alkaa laittamalla lähtösivun linkkikokoelmat prioriteettijonoon. Niin kauan kuin jono ei ole tyhjä, otetaan pois jonon kärjessä oleva linkkikokoelma, ja haetaan korkeintaan n siinä esiintyvää sivua. Haetut sivut jaetaan ryhmiin skeemoittain, jonka jälkeen yhdistellään samanlaisista skeemoista koostuvia ryhmiä.

Saadut ryhmät päivitetään nyt sivustomalliin, joka esitetään verkkona. Solmuina

on sivuryväät (joukko samanlaisia sivuja + sivujen skeemojen yhdiste), ja kaarina on näiden väliset suunnatut ryväslinkit. Ryhmä lisätään malliin joko uutena sivuryväänä, tai ryhmän sivut lisätään johonkin olemassaolevaan sivuryvääseen. Päätös tehdään laskemalla MDL-pituudet molemmille vaihtoehdoille, ja valitsemalla näistä se lyhyempi. Lopuksi kaikkien haettujen sivujen linkkikokoelmat lisätään prioriteettijonoon, ja algoritmin toiminta jatkuu.

MDL-periaatteen näkökulmasta kiinnostavaa on kuinka koodipituudet määräytyvät edellä esitetyssä algoritmissa. Koodattava sivustomalli M koostuu sivuryväistä $C_1, C_2, ..., C_n$. Sivuryväs C_i taas koostuu sivuista ja skeemasta. Malli koodataan ilmoittamalla sen sivuryväät, eli $L(M) = L(C_1) + L(C_2) + ... + L(C_n)$. Sivuryväs taas koodataan ilmoittamalla sen skeeman kaikki DOM-polut: $L(C_i) = L(path_1) + L(path_2) + ... + L(path_m)$.

Aineistona toimii ryväiden sivut $D = \{p_1, p_2, ..., p_k\}$. Yhden sivun esitys koostuu sen linkkikokoelmien DOM-poluista ja URL-osoitteista, esim: $p = \{path_1(url_{11}, url_{12}), path_3(url_{31}), path_4(url_{41}, url_{42})\}$. Jos esimerkkisivu p kuuluu ryvääseen $C_1 = \{path_1, path_2, path_3\}$, voidaan sivu koodata ryvään avulla seuraavasti: $enc(p|C_1) = enc(1) \cdot enc(\{url_{11}, url_{12}\}) \cdot enc(2) \cdot enc(\{\}) \cdot enc(3) \cdot enc(\{url_{31}\}) \cdot enc(path_4) \cdot enc(\{url_{41}, url_{42}\})$, missä $\cdot$ on katenaatio-operaatio. Jokaisen linkkikokoelman kohdalla ensin ilmoitetaan polku, sitten kokoelmassa esiintyvät URL-osoitteet. Jos polku löytyy apuna käytettävästä ryväästä, voidaan koodata vain sen indeksi. Muussa tapauksessa joudutaan eksplisiittisesti ilmoittamaan koko polku, kuten $path_4$ tapauksessa. Sivun koodauksessa joudutaan myös maksamaan sellaisista kokoelmista, jotka eivät esiinny sivulla mutta esiintyvät ryväässä. Esimerkki tällaisesta on $path_2$.

Koko aineiston koodipituudeksi saadaan siis jotain seuraavanlaista, sen mukaan mihin ryvääseen kukin sivu kuuluu: $L(D|M) = L(p_1|C_1) + L(p_2|C_1) + L(p_3|C_2) + ... + L(p_k|C_n)$.

MDL-pituuksia ei lasketa menetelmässä kovinkaan uskollisesti, vaan jokaiselle koodattavalle termille annetaan painoarvo väliltä $[0, 1]$ sen mukaan, mitä ollaan heuristisesti todettu hyväksi. Esim. $L(enc(url_i)) = 1$, $L(enc(path_i)) = 1$ ja $L(enc(n)) = 0.8$. MDL-periaatetta voidaan tulkita monella tapaa, eikä sitä aina sovelleta teorian kannalta eheästi vaan sen mukaan, mikä toimii käytännössä.

4.2 Optimointimenetelmien apuna

MDL-periaatetta voidaan käyttää apuna erilaisissa optimointimenetelmissä kuten geneettisissä algoritmeissa tai simuloidussa jäähdytyksessä. Esimerkiksi Li ja kumppanit esittelevät [LiA96] kuinka korpuksesta, eli kokoelmasta kirjoitetun kielen tekstejä, voidaan verbejä ja substantiiveja jakaa ryväisiin. Nämä ryväät muodostavat tesauruksen eli synonyymisanakirjan. Menetelmässä käytetään simuloitua jäähdytystä (engl. *simulated annealing)*, jonka energiafunktion arvoina on MDL-periaatteella lasketut pituudet.

Keller ja Lutz [KeL97, KeL05] esittelevät geneettisen algoritmin, jolla voidaan oppia stokastinen yhteydetön kielioppi (engl. *stochastic context-free grammar)* annettujen positiivisten esimerkkien avulla. MDL-periaate astuu peliin sopivuusfunktion (engl. *fitness function)* kautta.

Stokastiset yhteydettömät kieliopit esiintyvät monessa kieleen liittyvässä käytännön sovelluksessa, kuten esimerkiksi puheen tunnistuksessa, luonnollisen kielen prosessoinnissa, bioinformatiikassa ja tekstintunnistuksessa.

Yhteydetön kielioppi koostuu sijoitussäännöistä, joita soveltamalla saadaan aikaiseksi merkkijonoja s. Yhteydettömän kieliopin G tuottama kieli $L(G)$ koostuu kaikista niistä merkkijonoista, joita voidaan saada aikaiseksi soveltamalla kieliopin sääntöjä [Sip06]. Alla on esimerkki kielioppi:

$$
\begin{aligned}
&S \to AB \\
&A \to a \\
&A \to aB \\
&B \to b \\
&B \to bB
\end{aligned}
$$

Merkkijono ab voidaan tuottaa annetussa kieliopissa seuraavasti $S \implies AB \implies aB \implies ab$. Sijoitukset voidaan esittää myös puumuodossa, jolloin saadaan jäsennyspuu (engl. *parse tree)*. Yhdellä merkkijonolla voi olla useampi jäsennyspuu [Sip06].

Stokastinen yhteydetön kielioppi, lyhyemmin SCFG on kuten tavallinen yhteydetön kielioppi, mutta sääntöihin eli produktioihin liitetään todennäköisyyksiä väliltä $[0, 1]$. Alla on esimerkki SCFG:

$$
\begin{aligned}
S &\to AB \quad (1.0)\\
A &\to a \quad (0.7)\\
A &\to aB \quad (0.3)\\
B &\to b \quad (0.5)\\
B &\to bB \quad (0.5)
\end{aligned}
$$

SCFG on konsistentti, kun kaikkien samaa muuttujaa laajentavien sääntöjen todennäköisyyksien summa on tasan yksi. SCFG:n tuottama kieli $L(G)$ on sama kuin sitä vastaavan yhteydettömän kieliopin tuottama kieli. Merkkijonon $s \in L(G)$ todennäköisyys on $P_G(s) = \sum_k P(x_k)$, missä x_k ovat merkkijonon eri jäsennyspuita. Yhden jäsennyspuun todennäköisyys $P(x_k)$ on yhtä kuin siinä käytettyjen sääntöjen todennäköisyyksien tulo. Yllä esitetyssä esimerkkikieliopissa todennäköisyyksiksi saadaan mm.

$$
\begin{aligned}
P_G(ab) &= P(S \to AB) \cdot P(A \to a) \cdot P(B \to b) = 0.35.\\
P_G(abb) &= P(S \to AB) \cdot P(A \to a) \cdot P(B \to bB) \cdot P(B \to b) +\\
&\qquad P(S \to AB) \cdot P(A \to aB) \cdot P(B \to b) \cdot P(B \to b) = 0.175 + 0.075 = 0.25.
\end{aligned}
$$

Geneettisessä algoritmissaan Keller ja Lutz eivät kuitenkaan suoraan manipuloi SCFG-kielioppeja. Sen sijaan käytetään ennakollisia painotettuja kielioppeja (engl. *biased weight grammars, BWG)*, joissa todennäköisyyksien sijaan sääntöihin liitetään kaksi arvoa: ennakkoarvo ja painoarvo. Ennakkoarvo on muuttumaton arvo, jolla voidaan ennalta määrittää suosiollisempia sääntöjä. Tässä tapauksessa suositaan yksinkertaisempia sääntöjä monimutkaisten sijaan. BWG ja SCFG ovat ekvivalentit mallit, ja BWG saadaan muunnettua SCFG:ksi laskemalla yksinkertaisella kaavalla [KeL97] todennäköisyydet SCFG:n säännöille.

Korpus C on äärellinen otos kielestä L, jossa jokaiseen merkkijonoon $s \in C$ liittyy lisäksi luonnollinen luku f_s, joka kuvaa merkkijonon esiintymisfrekvenssiä. Korpuksen kokoa merkitään $N_C = \sum_{s \in C} f_s$. Merkkijonon suhteellinen esiintymisfrekvenssi on $p_s = f_s \,/\, N_C$.

Olkoon annettuna korpus C. Tehtävänä on löytää sellainen SCFG, joka määrittelee annetun korpuksen mahdollisimman tarkasti, mutta joka myös yleistyy sopivasti kattamaan koko populaatiota eli kieltä, josta C alunperin otettiin. Ensimmäisen ehdon tarkkuus voidaan kvantifioida seuraavasti:

$$
P(C|G) = \frac{N_C!}{\prod_{s \in C} f_s!} \prod_{s \in C} P_G(s)^{f_s}
$$

eli korpuksen todennäköisyys annetulla kieliopilla G. Paras kielioppi (suurimman uskottavuuden estimaatti) on tällöin kielioppi $\hat{G}$, joka maksimoi todennäköisyyden $P(C|G)$. Paras kielioppi $\hat{G}$ on sellainen, jonka säännöt tuottavat täsmälleen korpuksen merkkijonot todennäköisyyksillä, jotka vastaavat merkkijonojen suhteellisia frekvenssejä korpuksessa.

Huomioimalla ainoastaan kieliopin tarkkuus, paras kielioppi ylisovittuu annettuun aineistoon. Tällöin jälkimmäinen ehto, eli yleistys koko kieleen ei toteudu. Etsitään siis sen sijaan kielioppia, joka on todennäköisin annetun korpuksen valossa. Bayesin säännön nojalla voidaan laskea todennäköisyys kieliopille G ehdolla korpus C seuraavasti:

$$P(G|C) = \frac{P(G)P(C|G)}{P(C)}.$$

Kaavassa nimittäjä on vakio, joka voidaan unohtaa, kun tavoitteena on maksimoida funktion arvo. $P(G)$ on kieliopin G prioritodennäköisyys, ja $P(C|G)$ voidaan laskea kuten aikaisemmin. Prioritodennäköisyydet tullaan valitsemaan kieliopin monimutkaisuuden mukaan niin, että suositaan yksinkertaisia kielioppeja monimutkaisten sijaan.

Käytännössä $P(G|C)$ laskeminen on kovin työlästä. MDL-periaatteen avulla voidaan kuitenkin käyttää seuraavaa sopivuusfunktiota:

$$F(G) = \frac{K_C}{L(C|G) + L(G)}.$$

Yhtälön maksimointi vastaa nimittäjän minimointia. Tämä taas vastaa $P(G|C)$ maksimointia. Yhtälön osoittaja on korpuksesta riippuvainen vakio, joka normalisoi yhtälön arvon välille $[0, 1]$. Informaatioteorian nojalla osataan laskea, että

$$L(C|G) = -\log_2 P(C|G) = -\log_2 \left(\frac{N_C!}{\prod_{s \in C} f_s!} \right) - \sum_{s \in C} f_s \log_2(P_G(s)).$$

Ensimmäisen termin voi tiputtaa pois, sillä sen arvo ei riipu valitusta kieliopista. Todennäköisyyksiä $P(G)$ kieliopeille G ei tunneta, joten ei voida niiden avulla laskea pituuksia $L(G)$ suoraan kuten korpuksen tapauksessa. Sen sijaan laaditaan esitystapa kieliopeille G, ja lasketaan esityksen pituus. Esityksen pituudet muodostavat tällöin prioritodennäköisyysjakauman $P(G)$, sillä kuten informaatioteoriasta tiedetään, jokainen koodipituus on tosiasiassa todennäköisyys.

BWG:n säännöt koodataan muodossa (r, w), jossa r on sääntö ja w sen painoarvo. Ennakkoarvoja ei tarvitse koodata, koska ne voidaan päätellä säännöistä. Jos säännön painoarvo on nolla, se jätetään esityksestä pois. Painoarvo w esitetään

seuraavasti: ensin ilmoitetaan painoarvon w esityksen pituus ykkösillä, sen jälkeen käytetään välimerkkinä nollaa, ja lopuksi ilmoitetaan painoarvo w normaalilla binääriesityksellä. Pituudeksi siis saadaan $2\log_2(\log_2(w)) + 1 + \log_2(w)$.

Säännöt r koodataan niiden avulla määritellyn ennakkoarvon eli todennäköisyysjakauman $\hat{P}(r)$ avulla niin, että lyhyemmät säännöt saavat suurempia todennäköisyyksiä kuin pitkät. Siis lyhyemmät säännöt saavat myös lyhyemmät esityspituudet $-\log_2 \hat{P}(r)$. Kaava todennäköisyyden laskemiseksi löytyy lähteistä [KeL05, KeL97]. Siis BWG:n esityksen pituudeksi saadaan:

$$L(G) = \sum_{(r,w)\in G} (-\log_2(\hat{P}(r)) + \log_2(w) + 2\log_2(\log_2(w)) + 1).$$

Alla oleva geneettinen algoritmi käyttää äsken määriteltyä sopivuusfunktiota $F(G)$ peitekieliopin parametrien oppimiseen, eli BWG:n sääntöjen painoarvojen oppimiseen. Peitekielioppi on sellainen kielioppi, joka tuottaa ainakin korpuksen C merkkijonot. Lopussa saatu BWG voidaan muuntaa SCFG:ksi, ja ne säännöt, joiden todennäköisyys on 0, voidaan tiputtaa kieliopista pois.

Algoritmi 1 Geneettinen algoritmi peitekieliopin parametrien oppimiseen [KeL97]

Syöte: Korpus C

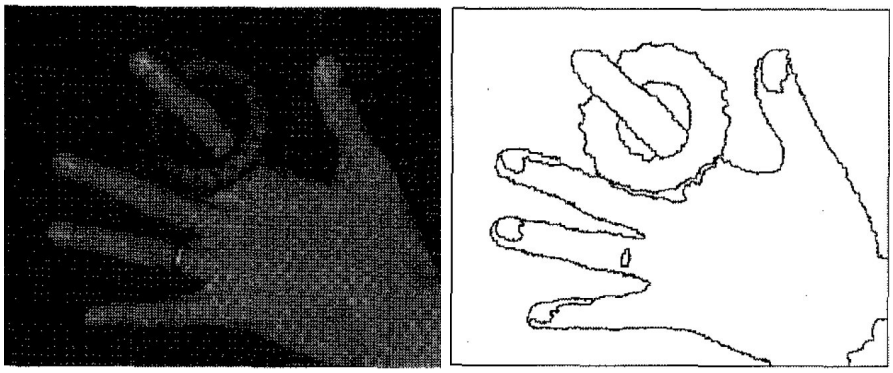
Tuloste: Painoarvot peitekieliopin G säännöille

1. Konstruoi peitekielioppi G.
2. Luo populaatio, jonka jokainen jäsen määrittää kieliopin G jokaisen säännön painoarvon. Valitse painoarvot mielivaltaisesti.
3. Valitse parhaat yksilöt sovitusfunktion avulla, ja risteytä sekä mutatoi niitä. Jatka tätä vaihetta kunnes riittävän hyvä yksilö (painoarvot) on löydetty, tai kunnes jokin ennalta asetettu yritysmäärä ylitetään.

4.3 Muita sovelluksia

MDL-periaatetta voidaan soveltaa myös eksoottisempiin ongelmiin.

Prosessilouhinnassa (engl. *process mining)* tavoitteena on automaattisesti louhia liiketoimintaprosessimalleja annetuista lokitiedoista. Organisaatio voi esimerkiksi tutkia, millaisista askeleista sen työntekijöiden tyypillinen työpäivä koostuu, jos työntekijöiden toiminnot on kirjattu lokeihin. Ongelman ratkaisemiseksi on ehdotettu



Kuva 6: Esimerkki kuvan segmentoinnista [ZhY96].

useampaa algoritmia, mutta algoritmien tuottamien prosessimallien vertailuun ei ole löydetty ongelmatonta yleiskäyttöistä mittaa. Calders ja kumppanit [CGP09] tarjoavat uutta mittaa, joka perustuu MDL-periaatteeseen. Laatimalla koodaustapa prosessimalleille ja lokitiedoille voidaan MDL-periaatteella verrata eri malleja keskenään.

Kuvan segmentoinnissa pyritään jakamaan digitaalinen kuva segmentteihin eli yhtenäisiin alueisiin värien, ääriviivojen ja tekstuurien perusteella [ShS01]. Tarkoituksena on tunnistaa objekteja ja ääriviivoja kuvasta, kuten esimerkiksi kuvassa 6. Segmentointi on suuressa roolissa konenäön (engl. *computer vision)* tutkimusta, ja sitä voidaan käytännössä soveltaa kasvontunnistuksessa, sormenjäljentunnistuksessa ja lääketieteellisessä kuvantamisessa (engl. *medical imaging)*. MDL-periaatetta on menestyksekkäästi käytetty apuna erilaisissa segmentointimenetelmissä, kuten Zhunin ja Yuillen menetelmässä [ZhY96], joka yhdistelee monien eri menetelmien hyviä puolia.

5 Yhteenveto

Lyhimmän kuvauspituuden periaate on tehokas työkalu tilastollisen mallin valintaan, kun vaarana on ylisovittuminen aineistoon. Tulkitsemalla mallien hypoteesit sopivasti todennäköisyysjakaumiksi, voidaan informaatioteorian tuloksin aineiston tiivistetty pituus laskea helposti. MDL-periaatteen mukaan paras hypoteesi ei välttämättä ole se, jonka avulla aineisto saavuttaa tiiviimmän pituudensa. Myös hypoteesin esityken pituus lasketaan, jonka seurauksena mallin monimutkaisuudesta rangaistaan Occamin partaveitsen -hengen mukaisesti.

Käytännössä MDL-periaatetta voidaan soveltaa erilaisiin koneoppimis- ja hahmontunnistustehtäviin. Tässä tutkielmassa esiteltiin muutama sovellus, joita yhdisti se, että niissä pyritään oppimaan sopiva malli aineiston avulla.

Tässä tutkielmassa esitetty kaksiosainen MDL-periaate on yksinkertaistettu versio modernista MDL-periaatteesta, joka on sekä teoreettisesti eheämpi että käytännössä tehokkaampi. Modernin MDL-periaatteen ymmärtäminen vaatii kuitenkin hieman laajempaa informaatioteorian ja tilastotieteen tuntemusta.

Tätä tutkielmaa tehdessäni huomasin, että kirjallisuudessa esitetään välillä hyvinkin poikkeavia näkemyksiä ja painotuksia siitä, mistä MDL-periaatteessa on kyse. Joidenkin mielestä se on vain tapa välttää ylisovittamista, kun taas toisten mielestä se on uraauurtava ja elegantti ratkaisu moniin mallin valinnan perustavanlaatuisiin ongelmiin.

Henkilökohtaisesti olen sitä mieltä, että kaksiosainen MDL-periaate on lähinnä tapa välttää ylisovittamista, joskaan en ole tämän tutkielman puitteissa perehtynyt tarkemmin MDL-periaatteen matemaattiseen pohjaan, tai MDL-periaatteen kaikkiin mahdollisiin sovelluskohteisiin. Olen kuitenkin vakuuttunut siitä, että informaatioteoreettisessa näkökulmassa on jotain hyvin vangitsevaa, ja siksi se toimii niin hyvin.

Lähteet

BFP06 Böhm, C., Faloutsos, C., Pan, J.-Y. ja Plant, C., Robust information-theoretic clustering. *Proc. of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006, sivut 65–75.

BRY98 Barron, A., Rissanen, J. ja Yu, B., The minimum description length principle in coding and modeling. *IEEE Transactions on Information Theory*, 44,6(1998), sivut 2743–2760.

CGP09 Calders, T., Günther, C., Pechenizkiy, M. ja Rozinat, A., Using minimum description length for process mining. *Proc. of the 2009 ACM symposium on Applied Computing*, 2009, sivut 1451–1455.

Cha66 Chaitin, G., On the lengths of programs for computing binary sequences. *Journal of the Association for Computing Machinery*, 13, sivut 547–569.

CMM05 Crescenzi, V., Merialdo, P. ja Missier, P., Clustering web pages based on their structure. *Data & Knowledge Engineering*, 54,3(2005), sivut 279–299.

CoT06 Cover, T. ja Thomas, J., *Elements of Information Theory*. John Wiley & Sons, toinen painos, 2006.

For00 Forster, M., Key concepts in model selection: performance and generalizability. *Journal of Mathematical Psychology*, 44, sivut 205–231.

GLV00 Gao, Q., Li, M. ja Vitányi, P., Applying MDL to learn best model granularity. *Artificial Intelligence*, 121, sivut 1–29.

Grü00 Grünwald, P., Model selection based on minimum description length. *Journal of Mathematical Psychology*, 44, sivut 133–170.

Grü05 Grünwald, P., A tutorial introduction to the minimum description length principle. Teoksessa *Advances in Minimum Description Length: Theory and Applications*, Grünwald, P., Myung, I. ja Pitt, M., toimittajat, MIT Press, 2005, sivut 3–79.

Grü07 Grünwald, P., *The Minimum Description Length Principle*. MIT Press, 2007.

Haw04 Hawkins, D., The problem of overfitting. *Journal of Chemical Information and Computer Sciences*, 44,1(2004), sivut 1–12.

HaY01 Hansen, M. ja Yu, B., Model selection and the principle of minimum description length. *Journal of the American Statistical Association*, 96,454(2001), sivut 746–774.

KeL97 Keller, B. ja Lutz, R., Evolving stochastic context-free grammars from examples using a minimum description length principle. *ICML-97 Workshop on automata induction grammatical inference and language acquisition*.

KeL05 Keller, B. ja Lutz, R., Evolutionary induction of stochastic context free grammars. *Pattern Recognition*, 38,9(2005), sivut 1393–1406.

KMU95 Kilpeläinen, P., Mannilla, H. ja Ukkonen, E., MDL learning of unions of simple pattern languages from positive examples. *Computational Learning Theory, Second European Conference, EuroCOLT '95*, 1995, sivut 252–260.

Kol65 Kolmogorov, A., Three approaches to the quantitative definition of information. *Problems of Information Transmission*, 1,1(1965), sivut 1–7.

Kra49 Kraft, L., A device for quantizing, grouping, and coding amplitude-modulated pulses. Väitöskirja, Massachusetts Institute of Technology. Dept. of Electrical Engineering, Yhdysvallat, 1949.

LiA96 Li, H. ja Abe, N., Clustering words with the MDL principle. *Proc. of the 16th conference on Computational linguistics COLING'96*, osa 1, 1996, sivut 4–9.

LiV08 Li, M. ja Vitanyi, P., *An introduction to Kolmogorov Complexity and its applications*. Springer, kolmas painos, 2008.

MBB08 Mendenhall, W., Beaver, R. ja Beaver, B., *Introduction to probability and statistics*. Brooks/Cole, 13. painos, 2008.

MNP06 Myung, J., Navarro, D. ja Pitt, M., Model selection by normalized maximum likelihood. *Journal of Mathematical Psychology*, 50, sivut 167–179.

Ris78 Rissanen, J., Modeling by the shortest data description. *Automatica*, 14, sivut 465–471.

Ris84 Rissanen, J., Universal coding, information, prediction and estimation. *IEEE Transactions on Information Theory*, 30, sivut 629–636.

Sip06 Sipser, M., *Introduction to the Theory of Computation*. Thomson Course Technology, toinen painos, 2006.

Sol64 Solomonoff, R., A formal theory of inductive inference, part 1 and part 2. *Information and Control*, 7, sivut 1–22, 224–254.

ShS01 Shapiro, L. ja Stockman, G., *Computer Vision*. Prentice-Hall, 2001.

WaD99 Wallace, C. S. ja Dowe, D. L., Minimum message length and Kolmogorov complexity. *The Computer Journal*, 42,4(1999), sivut 270–283.

ZhY96 Zhu, S. C. ja Yuille, A., Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18,9(1996), sivut 884–900.