Tilastotieteen pro gradu -tutkielma

Kotiloiden luokittelu - neuroverkkojen ja multinomiaalisen logistisen regression sovellus

Atte Lintilä

Jyväskylän yliopisto
Matematiikan ja tilastotieteen laitos
12. huhtikuuta 2014

Tiivistelmä

Lintilä Atte: *Kotiloiden luokittelu - neuroverkkojen ja multinomiaalisen logistisen regression sovellus*,
Tilastotieteen pro gradu -tutkielma, 28 s. + liitteitä 13 s., Jyväskylän yliopisto, Matematiikan ja tilastotieteen laitos, 12. huhtikuuta 2014.

Tilastollisella luokittelulla tarkoitetaan, että tilastoyksiköitä luokitellaan tietyn luokittelumallin avulla toisensa poissulkeviin ennalta tunnettuihin luokkiin. Tässä työssä luokittelulla tarkoitetaan kotilolajien tunnistamista niistä otetuista digitaalisista kuvista saatavien piirteiden avulla. Luokiteltava aineisto on saatu keräämällä kotiloita rantavesistöistä. Aineistossa on neljä lajia siten, että yksilöitä lajista *Bithynia tentaculata* on 292, *Myxas glutinosa* 229, *Physa acuta* 49 ja *Radix balthica* 24. Kunkin yksilön digitaalikuvasta on muodostettu 64 piirremuuttujaa, jotka kuvaavat mm. yksilön kokoa tai väriä. Luokittelussa laji on vasteena ja piirremuuttujat selittävinä muuttujina. Tässä työssä luokittelu toteutetaan neuroverkoilla ja multinomiaalisella logistisella regressiolla.

Työn päätavoitteena on löytää valituilla menetelmillä mahdollisimman hyvä luokittelumalli. Aluksi mallit sovitetaan siten, että kaikki selittävät muuttujat ovat malleissa mukana. Mallien toimivuutta pyritään parantamaan valitsemalla mahdollisimman hyvä selittävien muuttujien yhdistelmä ja säätämällä `R`-funktioissa olevia parametreja (mm. neuroverkon piiloyksiköiden lukumäärää). Varsinaisen päätavoitteen lisäksi työssä tutkitaan, kumpi menetelmä toimii kotiloiden luokittelussa paremmin, neuroverkot vai multinomiaalinen logistinen regressio. Tärkeinpänä mittarina luokittelumallien hyvyydelle on keskimääräinen luokitteluvirhe, joka pyritään saamaan mahdollisimman pieneksi. Lisäksi luokittelumallien hyvyyttä arvioidaan tarkastelemalla keskimääräisiä sekaannusmatriiseja.

Tutkielmassa kotiloaineiston luokittelussa saadut tulokset ovat hyviä. Kotilot saadaan luokiteltua oikein yli 90 %:n tarkkuudella eli luokitteluvirhe on alle 10 %, kun käytetään neuroverkkoa. Multinomiaalisella logistisella regressiolla saadut luokittelutulokset eivät ole aivan yhtä hyviä vaan luokitteluvirhe on yli 10 %. Keskimääräisistä sekaannusmatrsiiseista nähdään, että molemmilla menetelmillä lajit *Bithynia tentaculata* ja *Myxas glutinosa* luokittuvat hyvin, mutta lajit *Radix balthica* ja *Physa acuta* huonommin. Erityisesti *Physa acuta* luokittuu huonosti. Huonosti luokittuvien lajien otoskoko on huomattavasti pienempi kuin hyvin luokittuvien, ja on luultavaa, että näiden lajien luokittelutuloksia saataisiin parannettua kasvattamalla kyseisten lajien otoskokoa.

Avainsanat: kotilo, lajitunnistus, luokittelu, luokitteluvirhe, multinomiaalinen logistinen regressio, neuroverkko, sekaannusmatriisi.

Sisältö

1 Johdanto

Tässä tutkielmassa esitellään tilastollista luokittelua ja siihen liittyviä käsitteitä. Lisäksi esitellään kaksi erilaista luokittelumenetelmää, neuroverkot ja multinomiaalinen logistinen regressio. Tarkoituksena on selvittää, miten hyvin kotiloiden koneellinen lajitunnistus onnistuu edellä mainituilla luokittelumenetelmillä. Tässä työssä koneellisella lajitunnistuksella tarkoitetaan sitä, että kotiloista otetuista digitaalisista valokuvista saatujen piirteiden avulla yritetään tietokoneohjelman avulla päätellä, mihin lajiin kukin kotilo kuuluu. Tällä hetkellä tutkijat luokittelevat kotilot yksitellen käsin, joten koneellinen luokittelu nopeuttaisi luokittelutyötä. Ongelmana on löytää sellainen tilastollinen malli, joka luokittelisi kotilot oikein kuvista saatujen piirteiden avulla. Tämän tutkielman ensisijaisena tavoitteena on etsiä luokittelumalli, joka luokittelee kotilot oikein mahdollisimman hyvin.

Tutkielman toisena tarkoituksena on verrata neuroverkkoja ja multinomiaalista regressiota, niiden käyttökokemuksia ja sitä, saadaanko toisella näistä menetelmistä parempia tuloksia. Multinomiaalinen logistinen regressio kuuluu ns. perinteisiin tilastollisiin menetelmiin. Perinteisiä tilastollisia menetelmiä on verrattu neuroverkkoihin laajemmin mm. artikkelissa Paliwal & Kumar (2009). Kyseisessä artikkelissa on koottu yhteen tuloksia eri tutkimuksista, joissa on vertailtu neuroverkkoja ja perinteisiä menetelmiä. Artikkelissa todetaan, että neuroverkot näyttäisivät useammin toimivan perinteisiä menetelmiä paremmin kuin huonommin.

Kotiloiden koneellista luokittelua on tehty myös aikaisemmin. Artikkelissa Joutsijoki & Juhola (2012) on käytetty luokittelumenetelmänä tukivektorikonetta ja artikkelissa Ärje et al. (2013) on käytetty monia eri luokittelumenetelmiä (mm. naiivi bayes ja satunnainen metsä). Edellämainituissa artikkeleissa pääpaino on pohjaeläinten luokittelussa, mutta niissä on ollut mukana myös 2 eri kotilolajia *Bithynia tentaculata* ja *Myxas glutinosa*. Kyseiset kotilolajit ovat mukana myös tässä työssä käytetyssä aineistossa.

Tämän työn sisältö on seuraava: Luvussa 2 kuvataan aineiston keruutapa, muuttujat ja muuttujien muodostaminen. Lisäksi tässä luvussa esitellään tutkimusongelma ja sen ratkaisussa käytettävät käsitteet. Luvussa 3 esitellään toinen luokittelumenetelmistä eli multinomiaalinen logistinen regressio. Luvussa 4 kerrotaan neuroverkkojen historiasta ja esitellään neuroverkkoihin liittyviä käsitteitä. Luvussa 5 on varsinainen aineiston analysointi. Luvussa esitellään neuroverkoilla ja multinomiaalisella logistisella regressiolla saatuja luokittelutuloksia. Luvun lopussa kerrotaan, miten valituilla menetelmillä etsitään mahdollisimman hyvä luokittelumalli. Luvussa 6 on pohdintaa ja yhteenveto saaduista tuloksista.

2 Aineiston ja tutkimusongelman esittely

Tutkimusaineisto on tuotettu Jyväskylän yliopiston Bio- ja ympäristötieteiden laitoksella. Aineistonkeruu on osa ”Täpläravun vaikutukset suurten boreaalisten järvien litoraalieliöyhteisöihin” -projektia. Projektin tarkoituksena on tutkia täpläravun vaikutuksia rantavyöhykkeiden eliöyhteisöihin suomalaisissa suurjärvissä. Täpläravu on tuotu Suomeen Pohjois-Amerikasta 1960-luvulla ja tästä syystä sen vaikutuksia on haluttu tutkia. Projektin yhteydessä aineistoa on kerätty neljältä eri paikalta Päijänteeltä ja Etelä-Saimaalta. Havaintoja on tehty rantavesistä Ruolahdesta, Hirvisaaresta, Kansaanniemestä ja Päijätsalosta. Näistä Ruolahdessa ja Päijätsalossa on täplärapuja ja Hirvisaaressa ja Kansaanniemessä ei ole. Projektia varten on kerätty tietoja useista pohjaeläinlajeista, mutta tässä tutkielmassa keskitytään aineistosta siihen osaan, joka koskee kotiloita.

Kuva 1: Aineistossa olevat kotilolajit: *Bithynia tentaculata*, *Radix balthica*, *Physa acuta* ja *Myxas glutinosa*.

Aineisto on saatu keräämällä järven pohjasta pohjaeläinnäytteitä, joiden mukana tulleet kotilot muodostavat tutkittavan aineiston. Aineistossa on havaintoja neljästä eri kotilolajista. Otoskoko on yhteensä 594, joka jakautuu neljään lajiin siten, että havaintoja lajista *Bithynia tentaculata* on 292, *Radix balthica* 24, *Physa acuta* 49 ja *Myxas glutinosa* 229 (kuva 1).

Muuttujia aineistossa on 66. Näistä `Class`-muuttuja kertoo havainnon luokan, eli mihin lajiin kyseinen yksilö kuuluu. `Rapu`-muuttuja kertoo, elääkö täplärapuja siinä vesistössä, jossa kyseinen kotilo on elänyt. Loput 64 muuttujaa on saatu Suomen ympäristökeskuksessa otetuista digitaalisista kuvista ImageJ tietokoneohjelman avulla (Abràmoff et al., 2004). Nämä 64 piirrettä mittaavat mm. kotilon väriä ja kokoa (ks. liite A, Rasband (1997), Ärje et al. (2013)). Kyseiset 64 muuttujaa standardoidaan ennen analyysejä vähentämällä muuttujan arvosta kyseisen muuttujan keskiarvo ja jakamalla näin saatu arvo muuttujan keskihajonnalla. Standardointi tehdään, jotta neuroverkkomallia sovitettaessa kaikkia muuttujia kohdeltaisiin samanarvoisesti (Hastie et al., 2009).

Tutkimusongelmana on luokitella kotilot koneellisesti käyttäen hyväksi 64 piirremuuttujaa ja `rapu`-muuttujaa. Luokittelu halutaan tehdä koneellisesti, koska se olisi yksinkertaista ja vähän aikaa vievää. Tällä hetkellä kotiloiden luokittelu tehdään kä-

sin. Tilastollisena mallina luokittelussa käytetään sekä neuroverkkomallia että multinomiaalista logistista regressiota. Tarkasteltavana asiana on itse luokittelun lisäksi myös tutkia, kumpi toimii luokittelijana paremmin, multinomiaalinen logistinen regressio vai neuroverkko.

Parasta luokittelumallia etsiessä tärkein mallin hyvyyden mittari on luokitteluvirhe. Luokitteluvirheen laskemiseksi täytyy aineisto ensin jakaa opetus- ja testiaineistoksi. Opetusaineistoa käytetään mallin parametrien estimointiin. Kun luokittelumallin parametrit on estimoitu opetusaineiston avulla, testiaineisto luokitellaan sovitetun mallin avulla. Luokitteluvirhe on väärin luokiteltujen osuus kaikista luokitelluista. Mitä pienempi luokitteluvirhe on, sitä parempi on luokittelumalli. Jos luokittelisimme opetusaineiston saadulla mallilla, niin tällöin väärin luokiteltujen osuutta kutsutaan opetusvirheeksi. Opetusvirhe ei toimi hyvin mallin hyvyyden mittarina, koska sen avulla ei yleensä pystytä tekemään päätelmiä mallin yleistettävyydestä koko populaation tasolle. Tästä syystä aineisto on jaettava erillisiksi opetus- ja testiaineistoiksi.(Venables & Ripley, 2002)

Luokitteluvirhe kertoo, kuinka hyvin luokittelija toimii testiaineistossa. Monesti on myös hyödyllistä saada tietoa, minkälaisia virheitä luokittelumalli tekee tai onko esimerkiksi jokin tietty luokka, jonka luokittelija luokittelee erityisen hyvin tai huonosti. Luokkakohtaiset virheet kootaan sekaannusmatriisiksi (Ripley, 2007). Sekaannusmatriisin sarake kertoo mallin ennustaman luokan ja matriisin rivi kertoo oikean luokan. Matriisin rivillä i ja sarakkeessa j on niiden yksilöiden lukumäärä, jotka kuuluvat oikeasti luokkaan i ja malli luokittelee ne luokkaan j. Näin ollen matriisin diagonaalilla olevat luokittuvat oikein ja diagonaalin ulkopuoliset väärin.

Taulukko 1: Esimerkki sekaannusmatriisista.

		Mallin ennustama luokka		
		Luokka A	Luokka B	Luokka C
Oikea Luokka	Luokka A	43	5	0
	Luokka B	12	23	0
	Luokka C	0	0	50

Taulukossa 1 on keksitty esimerkki sekaannusmatriisista. Tämän matriisin perusteella voitaisiin päätellä, että luokittelumalli toimii hyvin luokan C osalta, koska malli on luokitellut kaikki 50 luokkaan C kuuluvaa havaintoa oikein eikä yksikään ole luokittunut väärin luokkaan C. Luokat A ja B puolestaan luokittuvat keskenään ristiin jonkin verran. Esimerkiksi 12 luokkaan B kuuluvaa havaintoa luokiteltiin virheellisesti luokkaan A.

3 Multinomiaalinen logistinen regressio

Logistinen regressio on tilastollinen malli aineistolle, jossa vastemuuttuja on dikotominen muuttuja. Luokittelutilanteessa tämä tarkoittaa, että aineistossa vastemuuttujalla on vain kaksi eri luokkaa. Logistisen regression yleistystä moniluokkaiseen tilanteeseen kutsutaan multinomiaaliseksi logistiseksi regressioksi. Multinomiaalinen logistinen regressio tunnetaan myös muilla nimillä, mm. multinomiaalinen logit regressio on eräs kirjallisuudessa käytetty nimi ja toinen esimerkki on softmax-regressio. Tämä nimi tulee mallissa käytetystä softmax-funktiosta. (Retherford & Choe, 1993)

Multinomiaalisessa logistisessa regressiossa vastemuuttujan tulee olla kategorinen muuttuja, joka voi saada vähintään kolme erilaista arvoa. Se sopii siis tilastolliseksi malliksi luokittelutilanteeseen, jossa luokkia on kolme tai enemmän. Vastemuuttujan arvon täytyy yksilöidä tilastoyksikön luokka täydellisesti, eli luokat ovat toisensa poissulkevia ja tietty tilastoyksikkö voi kuulua ainoastaan yhteen luokkaan. Mallin selittävinä muuttujina voi olla sekä jatkuvia että kategorisia muuttujia. (Retherford & Choe, 1993)

Oletetaan luokittelutilanne, jossa aineistossa on K luokkaa. Luokkien oletetaan olevan toisensa poissulkevia. Aluksi täytyy valita yksi luokka ns. vertailuluokaksi, johon muita verrataan. Vertailuluokan valinnalla ei tulosten kannalta ole väliä, koska luokat ovat kategorisia ja niiden järjestyksellä ei ole väliä. Valitaan nyt vertailuluokaksi luokka K. Lisäksi olkoon selittävien muuttujien vektori eli piirrevektori $\mathbf{x} = (x_1, ..., x_p)^T$, joka siis sisältää p erilaista piirrettä. Ottamalla nämä seikat huomioon malli saadaan seuraavaan muotoon

$$\begin{aligned}
\log \frac{P(G = 1|X = \mathbf{x})}{P(G = K|X = \mathbf{x})} &= \beta_{10} + \beta_1^T \mathbf{x} \\
\log \frac{P(G = 2|X = \mathbf{x})}{P(G = K|X = \mathbf{x})} &= \beta_{20} + \beta_2^T \mathbf{x} \\
&\vdots \\
\log \frac{P(G = K-1|X = \mathbf{x})}{P(G = K|X = \mathbf{x})} &= \beta_{(K-1)0} + \beta_{K-1}^T \mathbf{x},
\end{aligned} \tag{1}$$

missä parametrit β ovat tuntemattomia estimoitavia parametreja ja $P(G = k|X = \mathbf{x})$, $k = 1, ..., K-1$, tarkoittaa todennäköisyyttä kuulua luokkaan k, kun piirrevektorin $\mathbf{x}$ arvot tunnetaan. Muunnosta $\log[P(G = k|X = \mathbf{x})/P(G = K|X = \mathbf{x})]$ kutsutaan logit-muunnokseksi. Malli siis saadaan yksilöityä muodostamalla $K-1$ kappaletta logit-muunnoksia, joissa jokaisessa on käytetty logaritmin sisällä olevassa jakolaskussa jakajana todennäköisyyttä kuulua luokkaan K. (Hastie et al., 2009)

Kaava (1) saadaan muokattua kaavassa (2) olevaan muotoon ratkaisemalla yhtälöistä todennäköisyydet kuulua kuhunkin luokkaa eli ratkaisemalla $P(G = k|X = \mathbf{x})$,

missä $k = 1, ..., K-1$. Lisäksi tulee ottaa huomioon, että eri luokkiin kuulumisen todennäköisyyksien tulee summautua ykköseksi. Tällöin

$$\begin{aligned} P(G = k|X = \mathbf{x}) &= \frac{\exp(\beta_{k0} + \beta_{K-1}^T \mathbf{x})}{1 + \sum_{l=1}^{K-1} \exp(\beta_{l0} + \beta_l^T \mathbf{x})}, k = 1, ..., K-1, \\ P(G = K|X = \mathbf{x}) &= \frac{1}{1 + \sum_{l=1}^{K-1} \exp(\beta_{l0} + \beta_l^T \mathbf{x})}. \end{aligned} \tag{2}$$

Multinomiaalinen logistinen regressiomalli sovitetaan yleensä käyttäen suurimman uskottavuuden menetelmää. Kun merkitään $\theta = \{\beta_{10}, \beta_1^T, ..., , \beta_{(K-1)0}, \beta_{K-1}^T\}$ ja tiettyyn luokkaan k kuulumisen todennäköisyys $p_k(\mathbf{x}; \theta) = P(G = k|X = \mathbf{x})$, saadaan mallin logaritminen uskottavuusfunktio muotoon

$$l(\theta) = \sum_{i=1}^{N} \log p_{gi}(x_i; \theta), \tag{3}$$

missä N on havaintojen määrä ja g_i kertoo, mihin luokkaan yksilö kuuluu. Malli saadaan sovitettua maksimoimalla kyseinen funktio parametrien θ suhteen. Maksimoinnissa voidaan käyttää hyväksi mm. Newtonin algoritmia (Hastie et al., 2009). Kun parametriestimaatit θ on saatu, niin uuden yksilön i luokittelua tehtäessä luokittelusääntö on seuraava. Jos

$$P(G = k|X = \mathbf{x}_i) \geq P(G = k'|X = \mathbf{x}_i)$$

kaikilla $k \neq k'$, niin yksilö luokitellaan luokkaan k.

4 Neuroverkot

4.1 Historia

Neuroverkot on keksitty aivojen mallintamista varten vuonna 1943. Silloin McCulloch ja Pitts kirjoittivat hermosolujen toiminnasta artikkelin, jossa he mallinsivat yksinkertaista neuroverkkoa virtapiirin avulla (McCulloch & Pitts, 1943). Vuonna 1949 tätä konseptia vietiin eteenpäin kirjassa Hebb (1949), jossa mm. osoitettiin, että ihmisen hermopolut vahvistuvat aina kun niitä käytetään.

Tietokoneiden kehittyessä näiden teorioiden alkeiden mallintaminen tuli mahdolliseksi. 1950-luvulla IBM:n tutkimuslaboratoriossa yritettiin ensimmäistä kertaa simuloida neuroverkkoa. Ensimmäinen yritys epäonnistui, mutta myöhemmät yritykset onnistuivat (Anderson & McNeill, 1992). 50-luvun lopulla Widrow ja Hoff kehittivät mallit, joita he kutsuivat ADALINE:ksi ja MADALINE:ksi (Widrow & Hoff, 1960). Näistä ADALINE oli yksinkertainen neuroverkko, jonka opetuksessa käytettiin gradient descent -menetelmää keskineliövirheiden minimointiin. MADALINE oli ensimmäinen neuroverkko, jota sovellettiin käytännön ongelmaan: sitä käytettiin eliminoimaan kaikuja puhelinlinjoilta.

1950-luvun lopussa Rosenblatt keksi perceptron-mallin, jota hän sovelsi McCulloch-hin ja Pittsin neuromallille (Rosenblatt, 1958). Tämä malli oli yhden piilokerroksen neuroverkko. Rosenblatt esitti myös backpropagation-algoritmin, jolla voi opettaa usean piilokerroksen neuroverkoja (Rosenblatt, 1962). Ensimmäiset algoritmin käyttöyritykset epäonnistuivat, mutta myöhemmin algoritmi on todettu hyväksi. Vuonna 1967 Cowan esitteli ensimmäistä kertaa sigmoid-aktivointifunktion neuroverkolle (Cowan, 1967).

Neuroverkkotutkimus pysähtyi hetkeksi vuonna 1969, koska Minsky ja Papert esittelivät yksinkertaisen neuroverkkomallin rajoitukset (Minsky & Papert, 1969). Kahden seuraavan vuosikymmenen aikana neuroverkkojen rajoituksista päästiin yli muutamien laajennusten avulla:

1. Useiden piiloyksiköiden yhdistelmät. Huomattiin, että piiloyksiköiden yhdistelmät voivat olla tehokkaampia kuin yksittäiset piiloyksiköt. Useat eri tutkijat kehittivät erilaisia opetusmenetelmiä suurille neuroverkoille. Näistä menetelmistä suurin osa perustui edelleen gradient descent -menetelmään.(Mehrotra et al., 1997)

2. Usein gradient descent -menetelmällä ei pystytä löytämään toimivaa ratkaisua tutkittavaan ongelmaan. Tämän ongelman ratkaisemiseksi on kehitetty satunnaisuuteen ja todennäköisyyksiin perustuvia menetelmiä sekä stokastisia menetelmiä. Hintonin ja Sejnowskin kehittämä Bolzmann machines on yksi esimerkki stokastisesta neuroverkosta. (Hinton & Sejnowski, 1986)

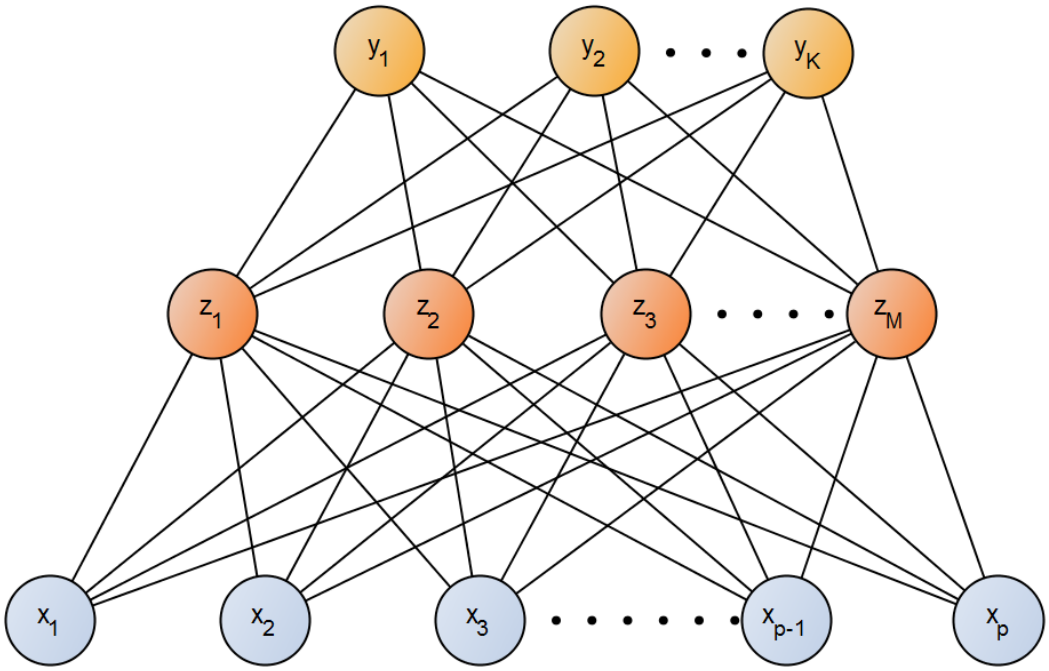
3. Hybrid systems kehitettiin (Sun, 1999). Siinä yhdistyy neuroverkot ja ei-konnektionistiset komponentit. Se kavensi kuilua symbolisten ja konnektionististen järjestelmien välillä. (Mehrotra et al., 1997)

Nykyään neuroverkkoja käytetään ennustamiseen ja luokitteluun mm. taloustieteessä, lääketieteessä ja erilaisissa teollisuuden suunnittelu- ja valmistusprosesseissa (Paliwal & Kumar, 2009).

4.2 Neuroverkkojen esittely

Sana neuroverkko käsittää ison määrän erilaisia tilastollisia malleja ja opetusmenetelmiä. Tässä osiossa perehdytään paljon käytettyyn yhden piilokerroksen backpropagation neuroverkkoon, ts. yhden kerroksen perceptroniin. Tätä neuroverkkoa voidaan käyttää luokittelumallina moniluokkaisessa tilanteessa.

Neuroverkko on siis saanut nimensä siitä, että sillä on alunperin yritetty mallintaa ihmisen aivojen ja hermoston toimintaa. Nimenä neuroverkko voikin tässä tapauksessa olla hieman harhaanjohtava. Kyseessä kuitenkin on vain epälineaarinen tilastollinen malli, jolla on samoja ominaisuuksia ja piirteitä kuin multinomiaalisella logistisella regressiolla.(Hastie et al., 2009)



Kuva 2: Yhden piilokerroksen neuroverkkodiagrammi.

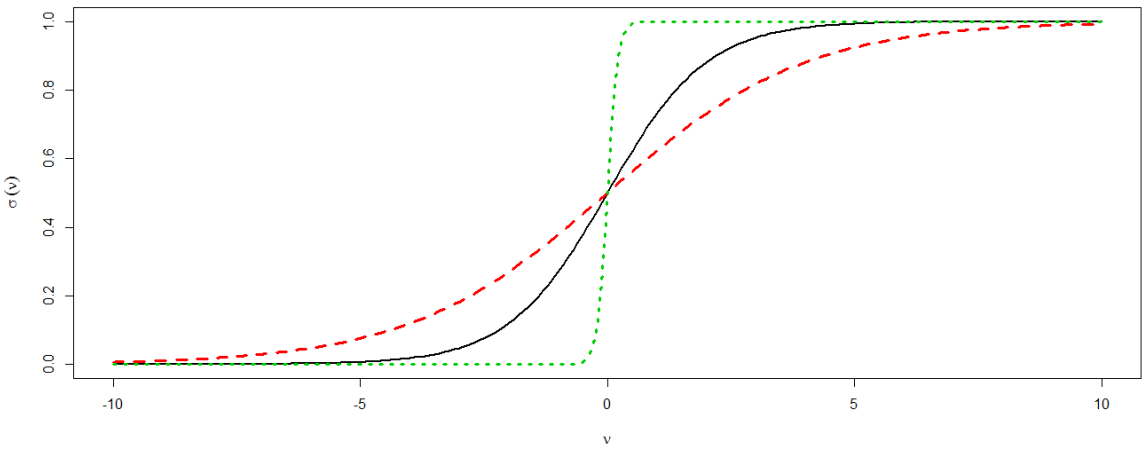
Neuroverkkomalli, jota käytetään luokitteluun moniluokkaisessa tilanteessa, esitetään usein neuroverkkodiagrammina, kuten kuvassa 2. Kuvassa ylimpänä olevaa vastemuuttujakerrosta kutsutaan ulostulokerrokseksi ja alimpana olevaa piirremuuttujakerrosta kutsutaan sisääntulokerrokseksi. Välikerrosta kutsutaan piilokerrokseksi. Piilokerroksia voi neuroverkkomallissa olla myös useampia kuin yksi. Teorian osalta nämä tapaukset sivuutetaan. Piilokerroksessa olevia yksiköitä, eli muuttujia

$z_m, m = 1, ..., M$, kutsutaan piiloyksiköiksi, koska niitä ei suoraan havaita. Yleisesti pätevää sääntöä piiloyksiköiden määrälle ei voida sanoa, mutta on parempi, jos piiloyksiköitä on liikaa kuin liian vähän. Eräänlaisena nyrkkisääntönä voidaan pitää, että piiloyksiköitä tulisi olla jotain 5-100 väliltä, riippuen selittävien muuttujien ja havaintojen määrästä (Hastie et al., 2009).

Oletetaan kuvan 2 kaltainen luokittelutilanne, eli aineistossa on K luokkaa. Tällöin vastemuuttujakerroksessa on K yksikköä, ja näistä k:s yksikkö mallintaa luokkaan k kuulumisen todennäköisyyttä. Aineistossa muuttuja $y_k = \{0, 1\}, k = 1, ..., K$, kertoo, kuuluuko yksilö luokkaan k, ts. y_k saa arvon 1, jos se kuuluu luokkaan k ja jos se ei kuulu luokkaan k, se saa arvon 0. Selittävien muuttujien vektori eli piirrevektori $\mathbf{x} = (x_1, ..., x_p)^T$ sisältää p erilaista piirrettä. Näiden lisäksi luokitteluun tarvitaan piirrevektoreista tehty muunnos $p_k(\mathbf{x}, \mathbf{\Psi})$. Tällöin

$$\begin{aligned} z_m &= \sigma(\alpha_{0m} + \alpha_m^T \mathbf{x}), m = 1, ..., M, \\ t_k &= \beta_{0k} + \beta_k^T \mathbf{z}, k = 1, ..., K, \\ p_k(\mathbf{x}, \mathbf{\Psi}) &= g_k(\mathbf{t}), k = 1, .., K, \end{aligned}$$

missä $\mathbf{z} = (z_1, z_2, ..., z_M)^T$, $\mathbf{t} = (t_1, t_2, ..., t_K)^T$ ja $\mathbf{\Psi}$ sisältää kaikki estimoitavat parametrit, ks. kaava (5). Aktivointifunktioksi $\sigma(\nu)$ valitaan yleensä sigmoid-funktio eli $\sigma(\nu) = 1/(1 + e^{-\nu})$; ks. kuva 3. Jos aktivointifunktio olisi identiteettifunktio, niin koko malli olisi vain lineaarinen malli piirremuuttujista, näin ollen neuroverkon voidaan ajatella olevan epälineaarinen yleistys lineaarisesta mallista.



Kuva 3: Sigmoid-aktivointifunktio. Kuvassa piirretty $\sigma(s\nu)$. Punaiselle "pitkä katkoviiva" s=0.5 ja vihreälle "lyhyt katkoviiva" s=10. Skaalaparametri s kontrolloi sitä, kuinka "aktiivinen" aktivointifunktio on.

Tavoitteena on mallintaa muuttujia y_k alkuperäisten piirteiden muunnosten $p_k(\mathbf{x}, \mathbf{\Psi})$ avulla, mikä vaatii tuntemattomien parametrien $\mathbf{\Psi}$ estimointia. Muunnos $p_k(\mathbf{x}, \mathbf{\Psi})$ saadaan siis funktiosta $g_k(\mathbf{t})$. Luokittelun yhteydessä sille käytetään muotoa

$$p_k(\mathbf{x}, \mathbf{\Psi}) = g_k(\mathbf{t}) = \frac{\exp(t_k)}{\sum_{l=1}^{K} \exp(t_l)} = \frac{\exp(\beta_{0k} + \beta_k^t \mathbf{z})}{\sum_{l=1}^{K} \exp(\beta_{0l} + \beta_l^T \mathbf{z})}. \quad (4)$$

Yllä olevassa kaavassa (4) olevaa funktiota $g_k(\mathbf{t})$ kutsutaan softmax-funktioksi. Saman tapaista funktiota käytetään myös multinomiaalisessa logistisessa regressiossa (2). Softmax-funktiosta saadaan arvoja väliltä [0,1] ja ne summautuvat ykköseen.

4.3 Neuroverkon sovitus

Neuroverkon sovittamisella tarkoitetaan tuntemattomien parametrien $\mathbf{\Psi}$ estimointia. Parametreja estimoitaessa aineisto jaetaan opetus- ja testiaineistoksi. Opetusaineistoa käytetään parametrien estimoimiseen ja testiaineistoa mallin hyvyyden testaamiseen. Parametriverktori $\mathbf{\Psi}$ sisältää seuraavat parametrit:

$$\begin{aligned} \{\alpha_{om}, \alpha_m; m = 1, 2, ..., M\} \text{ M(p+1) parametria,} \\ \{\beta_{0k}, \beta_k; k = 1, 2, ...K\} \text{ K(M+1) parametria.} \end{aligned} \quad (5)$$

Luokittelutilanteessa neuroverkon parametrit estimoidaan minimoimalla joko neliövirheiden summaa, kaava (6) tai minimoimalla ns. ristientropiaa, kaava (7):

$$R(\mathbf{\Psi}) = \sum_{k=1}^{K} \sum_{i=1}^{N} (y_{ik} - p_k(\mathbf{x}_i, \mathbf{\Psi}))^2 \quad (6)$$

$$R(\mathbf{\Psi}) = -\sum_{i=1}^{N} \sum_{k=1}^{K} y_{ik} \log(p_k(\mathbf{x}_i, \mathbf{\Psi})). \quad (7)$$

Käytettäessä softmax-funktiota mallintamiseen (4) ja ristientropiaa virhefunktiossa $R(\mathbf{\Psi})$ (7), kyseinen neuroverkkomalli on samanlainen piiloyksiköiden osalta kuin lineaarinen logistinen regressio, ks. kaava (3). Tällöin neuroverkkomallin kaikki parametrit voitaisiin estimoida suurimman uskottavuuden menetelmällä, kaava (8). Yleinen ratkaisu $R(\mathbf{\Psi})$:n minimoimiseksi on gradient descent -menetelmä, jota tässä yhteydessä kutsutaan backpropagation-algoritmiksi. Kun parametriestimaatit $\hat{\mathbf{\Psi}}$ on saatu, niin uuden yksilön x_i luokittelua tehtäessä luokittelusääntö on seuraava. Jos

$$p_k(\mathbf{x}_i, \hat{\mathbf{\Psi}}) \geq p_{k'}(\mathbf{x}_i, \hat{\mathbf{\Psi}})$$

kaikilla $k \neq k'$, niin yksilö luokitellaan luokkaan y_k.

Kaavasta (7) päästään uskottavuusfunktioon (8) seuraavien laskutoimitusten kautta:

$$\begin{aligned}
-R(\mathbf{\Psi}) &= \sum_{i=1}^{N}\sum_{k=1}^{K} y_{ik}\log(p_k(\mathbf{x}_i,\mathbf{\Psi})) = \sum_{i=1}^{N}\sum_{k=1}^{K}\log(p_k(\mathbf{x}_i,\mathbf{\Psi})^{y_{ik}}) \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K}\log(\pi_{ki}^{y_{ki}}) = \sum_{i=1}^{N}\log(\pi_{1i}^{y_{1i}}\pi_{2i}^{y_{2i}}\times\cdots\times\pi_{Ki}^{y_{Ki}}) \\
&= \log(\prod_{i=1}^{N}\pi_{1i}^{y_{1i}}\pi_{2i}^{y_{2i}}\times\cdots\times\pi_{Ki}^{y_{Ki}}) = l(\mathbf{\Psi},\mathbf{y},\mathbf{x}).
\end{aligned}$$

Näin siis uskottavuusfunktio parametreille $\mathbf{\Psi}$ on

$$L(\mathbf{\Psi},\mathbf{y},\mathbf{x}) = c\prod_{i=1}^{N}\pi_{1i}^{y_{1i}}\pi_{2i}^{y_{2i}}\times\cdots\times\pi_{Ki}^{y_{Ki}}, \tag{8}$$

missä

$$c = \frac{1}{y_{1i}!\times\cdots\times y_{Ki}!},$$

$$\pi_{ki} = p_k(\mathbf{x}_i,\mathbf{\Psi}) = \frac{\exp(\beta_{0k}+\beta_k^T\mathbf{z}_i)}{\sum_{l=1}^{K}\exp(\beta_{0l}+\beta_l^T\mathbf{z}_i)}$$

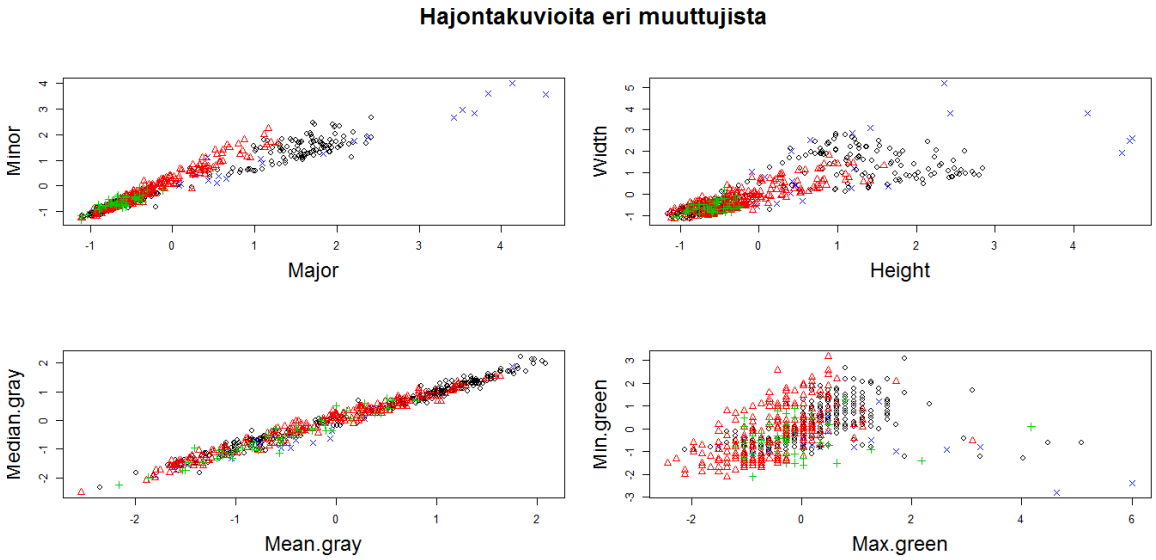
ja $y_{ki} = 1$, jos yksilö i kuuluu luokkaan k, muutoin $y_{ki} = 0$. Näin siis vain yksi termeistä eroaa nollasta.

5 Aineiston analysointi

Aineiston alkuanalyysissä ja kotiloiden koneellisessa luokittelussa käytetään hyväksi `R`-ohjelmointia (R Development Core Team, 2008) ja sille tehtyjä paketteja: `RWeka` (Hornik et al., 2009), jota hyödynnetään neuroverkkomallin sovituksessa, ja `nnet` (Venables & Ripley, 2002), jonka funktioita hyödynnetään sekä neuroverkon sovituksessa että multinomiaalisen logistisen regression sovituksessa. Lisäksi multinomiaalinen logistinen regressio tehdään `VGAM`-paketin (Yee, 2013) `vglm`-funktion avulla. Alkuanalyyseihin ja luokitteluun käytetty `R`-koodi on liitteessä B.

Aineiston analyysi aloitetaan standardoimalla jatkuvat muuttujat. Standardointi tehdään vähentämällä jokaisesta muuttujan arvosta kyseisen muuttujan keskiarvo ja jakamalla näin saatu luku muuttujan keskihajonnalla. Aineiston luokittelu tehdään sekä neuroverkkomallilla että multinomiaalisella logistisella regressiolla. Eri mallien tuloksia verrataan toisiinsa, ja vertailujen avulla etsitään mahdollisimman hyvä luokittelumalli. Ennen varsinaista luokittelua aineistosta piirretään hajontakuvioita. Hajontakuvioiden perusteella yritetään päätellä, löytyisikö muuttujia, jotka erottelisivat luokkia mahdollisimman hyvin. Yhtään erityisen hyvin erottelevaa muuttujaa ei kuvioiden perusteella löydy.

Kuvassa 4 on esimerkiksi neljä hajontakuviota, joista ylemmissä kuvioissa olevat muuttujat mittaavat eri tavoin kotilon kokoa, ja alemmissa kuvioissa olevat mittaavat kotilon väriä. Kuvioista nähdään, että täysin selkeitä erillisiä lajikohtaisia ryhmiä kuvioihin ei muodostu. Lisäksi kuvioista voidaan nähdä vahva korrelaatio eri muuttujien välillä. Tätä tietoa voidaan käyttää hyväksi, kun mietitään, mitä selittäviä muuttujia malliin kannattaa ottaa mukaan. Voisi olla mielekästä ottaa malliin vain yksi kokoa mittaava muuttuja, koska usean kokoa mittaavan muuttujan lisääminen ei välttämättä tuota lisäinformaatiota malliin, mutta lisää estimoitavien parametrien määrää. Kuvassa 4 on ainoastaan 8 eri muuttujaa, mutta vahvaa korrelaatiota on havaittavissa myös muissa muuttujissa (ks. liite C). Esimerkiksi muuttujat `Median.Blue` (sinisen mediaani) ja `Mean.Blue` (sinisen keskiarvo) mittaavat molemmat keskimääräistä sinisen värin määrää kotilossa. Näiden muuttujien välillä on vahva korrelaatio (n. 0.99) ja voisi olla mielekästä jättää ainakin toinen näistä pois selittävistä muuttujista.



Kuva 4: Neljä eri hajontakuviota. Eri lajit on piirretty eri väreillä siten, että *Bithynia tentaculata* on piirretty mustalla (ympyrät), *Radix balthica* sinisellä (×-merkit), *Myxas glutinosa* punaisella (kolmiot) ja *Physa acuta* vihreällä (+-merkit) värillä.

5.1 Aineiston luokittelu neuroverkolla

Neuroverkkoluokittelijan luomisessa käytetään hyväksi `nnet` ja `RWeka` R-paketteja. Yhden piilokerroksen neuroverkko voidaan estimoida nnet-paketin avulla. Tällaisen neuroverkon sovituksessa käytetään paketin nnet-funktiota, joka käyttää mallin parametrien estimoinnissa BFGS-menetelmää. Se on hieman samanlainen optimointialgoritmi kuin edellä mainittu gradient descent -menetelmä. Käytettäessä nnet-funktiota täytyy käyttäjän itse valita optimaalinen piiloyksiköiden lukumäärä. RWeka-paketin avulla voidaan taas muodostaa neuroverkkoja, joissa on useampi piilokerros. Funktion `make\_weka\_classifier` avulla muodostetaan usean piilokerroksen neuroverkkoluokittelija. Funktio määrittelee automaattisesti sopivan piilokerrosten ja piiloyksiköiden lukumäärän (piilokerrosten lukumäärä voi olla myös 1). Mallin parametrien estimoinnissa kyseinen funktio käyttää hyväksi backpropagation-algoritmin muokattua versiota (Gradient Descent with Momentum and Adaptive Learning Rate Backpropagation, ks. esim. Joutsijoki et al. (2014)).

Luokittelu aloitetaan jakamalla aineisto opetus- ja testiaineistoksi. Opetusaineistoksi valittiin satunnaisesti puolet havainnoista ja loput jätetään testiaineistoksi. Ensimmäisenä opetusaineistosta estimoidaan luokittelumalliin liittyvät parametrit, ts. opetetaan neuroverkko. Seuraavaksi opetettua neuroverkkoa käytetään testiaineiston luokitteluun. Lopuksi testiaineiston luokittelutuloksista lasketaan, kuinka suuri osa aineistosta luokiteltiin väärin. Tätä osuutta kutsutaan luokitteluvirheeksi. Luokitteluvirhettä käytetään luokittelijan hyvyyden tarkasteluun. Neuroverkkoluokittelijan luokitteluvirheen epävarmuutta arvioidaan toistamalla edellä mainitut vaiheet 100 kertaa ja taulukoidaan näin saadut luokitteluvirheet. Aluksi luokittelumallin

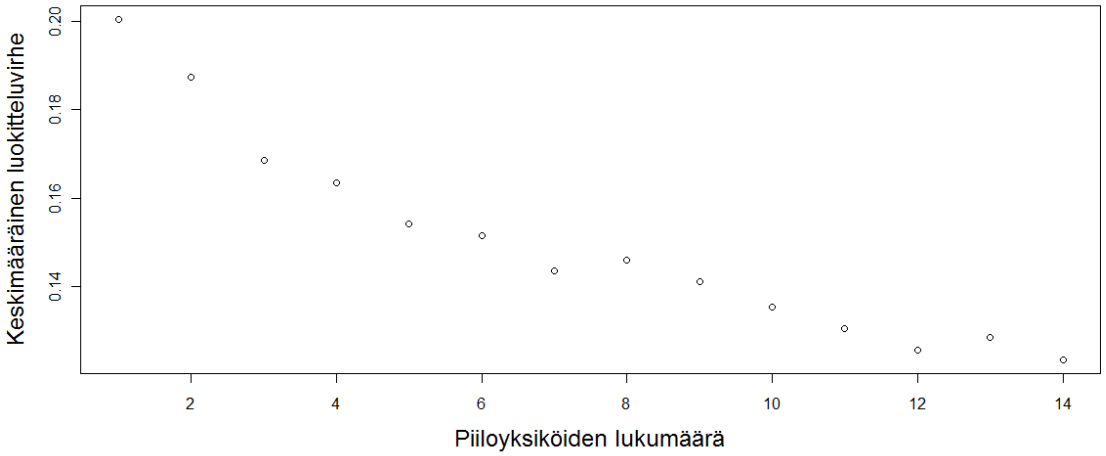
selittävinä muuttujina käytetään kaikkia 65 muuttujaa. Parasta luokittelumallia etsiessä (Luku 5.3) mietitään myös, mitkä muuttujat tulisi pitää mukana mallissa ja mitkä voisi jättää pois.

Käytettäessä `nnet`-funktiota täytyy piiloyksiköiden lukumäärä itse määrittää. Piiloyksiköiden määrän tulee olla jotain väliltä 1-14, koska jos piiloyksiköitä on 15 tai enemmän, `nnet`-funktio ei enää toimi tällä aineistolla. Tämä johtuu siitä, että estimoitavia parametreja on liikaa suhteessa opetusaineiston kokoon. Jos selittävien muuttujien määrää lasketaan, voidaan samalla piiloyksiköiden määrää nostaa. Piiloyksiköiden määrä päätetään siten, että suoritetaan neuroverkkomallin sovitus ja testiaineiston luokittelu 14 kertaa samalla opetus- ja testiaineistolla seuraavasti. Ensimmäisellä kerralla piiloyksiköiden määrä on 1 ja jokaisella kerralla sitä kasvatetaan yhdellä. Jokaiselta luokittelukerralta lasketaan luokitteluvirhe. Lopuksi katsotaan, millä määrällä piiloyksiköitä luokitteluvirhe oli pienin. Edellä mainitut vaiheet toistetaan 100 kertaa ja katsotaan, millä piiloyksiköiden määrällä saadaan paras luokittelumalli useimmiten. Taulukosta 2 nähdään, että kaikkien selittävien muuttujien ollessa mukana mallissa, parhaaksi piiloyksiköiden määräksi saadaan 14.

Taulukko 2: Sadasta eri luokittelukerrasta saatujen piiloyksiköiden lukumäärä siten, että alemmalla rivillä on niiden luokittelukertojen määrä, jolloin ko. lukumäärä on ollut paras kaikista.

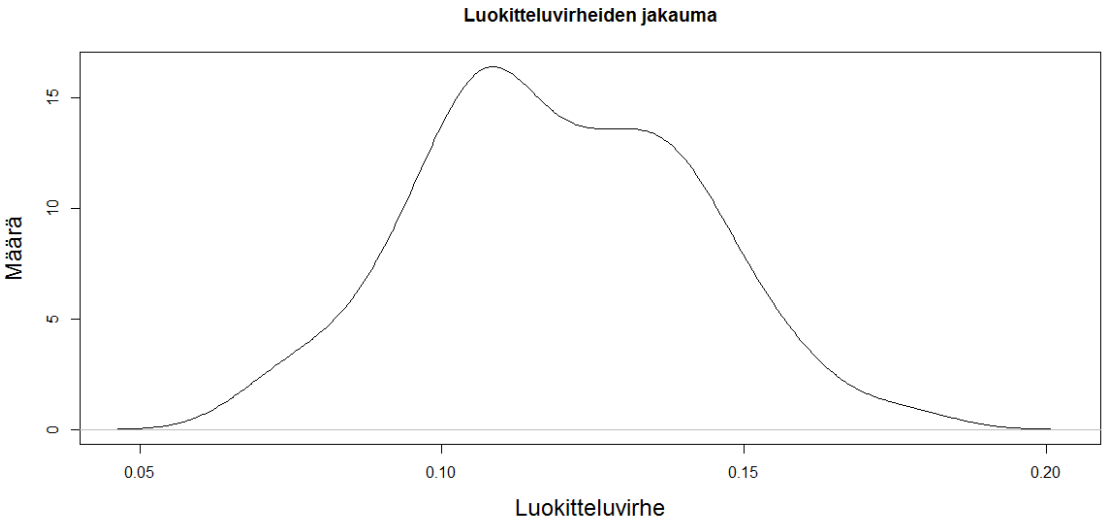
Piiloyksiköt	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**Yht.**
Paras(lkm)	0	0	0	2	5	3	2	5	8	10	14	11	14	26	100

Piiloyksiköiden määrää on tutkittu lisäksi toistamalla luokittelu 100 kertaa kullakin piiloyksiköiden määrällä ja laskemalla jokaiselta määrältä keskimääräiset luokitteluvirheet. Kuvasta 5 nähdään luokittelujen tulokset. Myös nyt paras valinta on mahdollisimman paljon piiloyksiköitä eli 14. Kuvasta nähdään, että piiloyksiköiden määrän valinnalla on melko suuri merkitys. Näyttää siltä, että piiloyksiköiden määrän kasvaessa luokitteluvirhe pienenee.



Kuva 5: Eri piiloyksiköiden lukumäärillä saadut keskimääräiset luokitteluvirheet.

Kuvassa 6 on toistamalla saatujen luokitteluvirheiden jakauma, kun luokittelu on tehty yhden piilokerroksen neuroverkolla, jossa on 14 piiloyksikköä. Luokitteluvirheiden keskiarvo on 0.119 ja keskihajonta 0.023.



Kuva 6: Toistamalla saatujen luokitteluvirheiden jakauma yhden piilokerroksen neuroverkolla.

Yksittäisen luokittelukerran hyvyyden tarkastelussa käytetään luokitteluvirheen lisäksi apuna sekaannusmatriisia, joka kuvaa sitä, kuinka suuri osa tiettyyn lajiin kuuluvista yksilöistä luokitellaan väärin. Taulukossa 3 esitellään erään luokittelun tulokset. Kyseinen luokittelu on tehty käyttäen yhden piilokerroksen neuroverkkoa. Taulukon yksittäisessä solussa on tieto siitä, kuinka monta yksilöä luokittui tietystä lajista tiettyyn luokkaan. Diagonaalilla on siis oikein luokitellut ja diagonaalin ulkopuolella virheellisesti luokitellut.

Taulukko 3: Yhden piilokerroksen neuroverkon sekaannusmatriisi yhdeltä luokittelukerralta testiaineistolla.

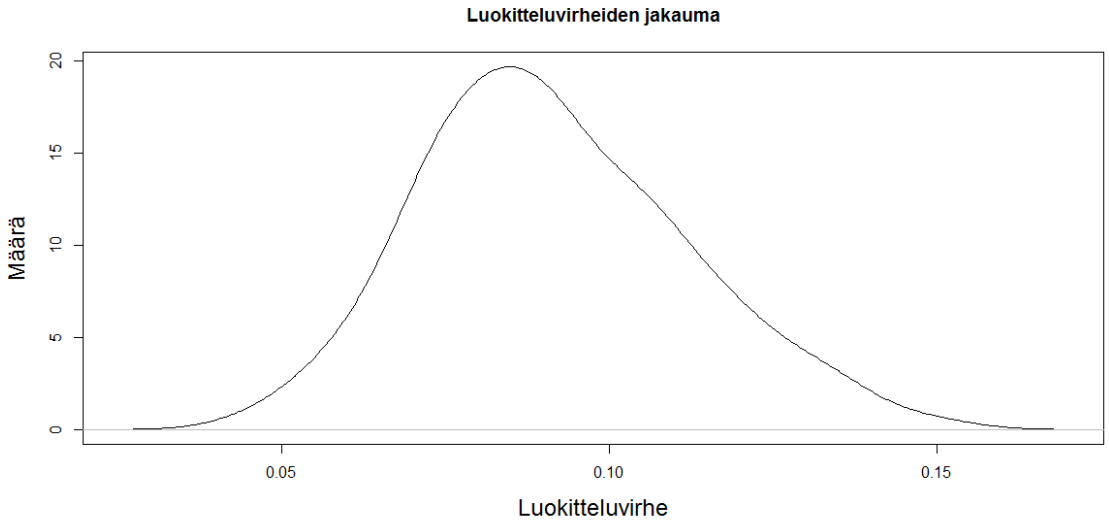
	Bithynia tentaculata	*Myxas glutinosa*	*Physa acuta*	*Radix balthica*	Yhteensä
Bithynia tentaculata	132	7	5	6	150
Myxas glutinosa	7	97	3	2	109
Physa acuta	4	0	23	1	28
Radix balthica	2	1	0	7	10

Taulukosta 3 nähdään ainoastaan yhden satunnaisen luokittelukerran tulokset. Tästä ei yleensä pystytä päättelemään, mitä virheitä luokittelija yleisesti tekee. Jotta saataisiin parempi kuva luokittelijan tekemien virheiden laadusta, niin lasketaan vielä lisäksi keskimääräinen sekaannusmatriisi, ks. esim. Joutsijoki et al. (2014). Keskimääräinen sekaannusmatriisi saadaan toistamalla luokittelu 100 kertaa, kuten edellä on kerrottu, ja laskemalla jokaiselta luokittelukerralta sekaannusmatriisi. Kun matriisit on laskettu, summataan matriisit yhteen ja jaetaan kyseinen matriisi toistojen lukumäärällä eli sadalla. Jatkossa tutkitaan yhden kerran sekaannusmatriisin sijaan ainoastaan keskimääräisiä sekaannusmatriiseja.

Taulukko 4: Yhden piilokerroksen neuroverkon keskimääräinen sekaannusmatriisi.

	Bithynia tentaculata	*Myxas glutinosa*	*Physa acuta*	*Radix balthica*	Yhteensä
Bithynia tentaculata	136.1	4.3	3.8	2.4	146.6
Myxas glutinosa	7.2	103.0	2.3	1.6	114.1
Physa acuta	3.3	2.4	18.2	0.4	24.3
Radix balthica	6.1	0.9	0.7	4.3	12.0

Taulukosta 4 huomataan, että lajit *Bithynia tentaculata* ja *Myxas glutinosa* neuroverkkoluokittelija luokittelee hyvin. *Bithynia tentaculata* luokittuu oikein n. 93 %:n tarkkuudella ja *Myxas glutinosa* 90 %:n tarkkuudella. Lajin *Physa acuta* neuroverkkomalli luokittelee hieman huonommin kuin edellä mainitut, oikein luokittuu keskimäärin n. 75 % lajin kotiloista. Lajin *Radix balthica* luokittelu onnistuu huonosti. *Radix balthica* näyttäisi luokittuvan oikein vain 36 %:n tarkkuudella. Taulukosta huomataan myös se, että nimenomaan kahden viimeksi mainitun lajin havaintomäärät ovat huomattavasti pienemmät kuin kahden muun.



Kuva 7: Toistamalla saatujen luokitteluvirheiden jakauma usean piilokerroksen neuroverkolla.

Kuvassa 7 on toistamalla saatujen luokitteluvirheiden jakauma, kun luokittelumallina on usean piilokerroksen neuroverkko. Luokitteluvirheiden keskiarvo on 0.092 ja keskihajonta 0.020, joten usean piilokerroksen neuroverkkomallilla saadut tulokset ovat hieman parempia kuin yhden piilokerroksen neuroverkkomallilla. Usean piilokerroksen neuroverkolla saatu keskimääräinen luokitteluvirhe on 0.027 yksikköä pienempi kuin yhden piilokerroksen neuroverkolla. Tässä vaiheessa ei kuitenkaan vielä tehdä lopullista päätöstä menetelmien paremmuudesta, sillä seuraavaksi tutkitaan tarkemmin mallin selittäviä muuttujia ja `R`-funktioissa olevien säätöparametrien tarpeellisuutta.

Taulukosta 5 nähdään, että usean piilokerroksen neuroverkko tekee samanlaisia virheitä kuin yhden piilokerroksen neuroverkkokin. Lajit *Bithynia tentaculata* ja *Myxas glutinosa* usean piilokerroksen neuroverkkoluokittelija luokittelee hyvin, kun taas lajit *Physa acuta* ja *Radix balthica* huonommin. Erityisesti *Radix balthica* luokittuu todella huonosti. *Bithynia tentaculata* luokittuu oikein n. 96 %:n, *Myxas glutinosa* 95 %:n, *Physa acuta* 78 %:n ja *Radix balthica* 19 %:n tarkkuudella.

Taulukko 5: Usean piilokerroksen neuroverkon keskimääräinen sekaannusmatriisi.

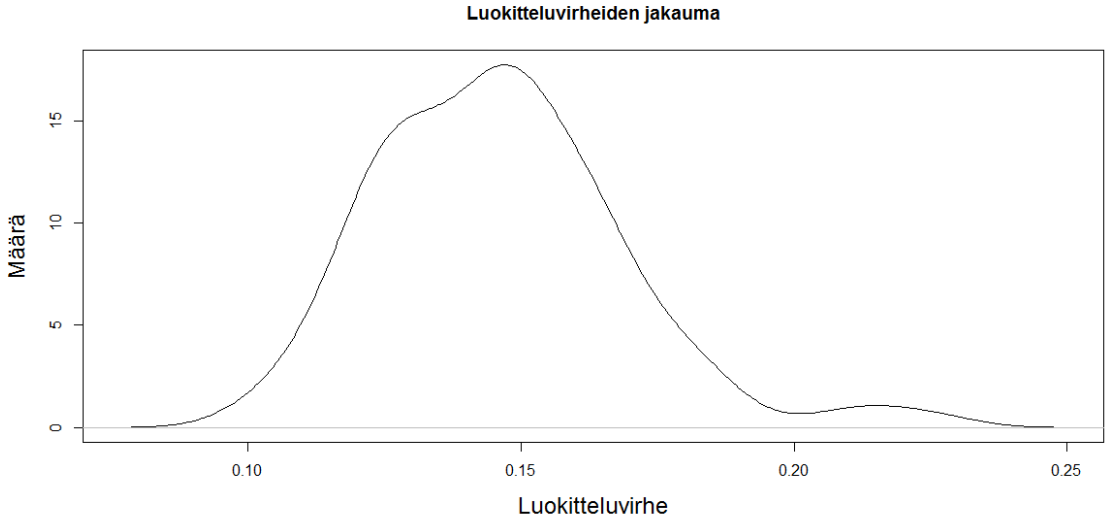
	Bithynia tentaculata	*Myxas glutinosa*	*Physa acuta*	*Radix balthica*	Yhteensä
Bithynia tentaculata	140.2	3.5	2.6	0.4	146.7
Myxas glutinosa	4.5	108.4	0.8	0.3	114.0
Physa acuta	3.5	1.9	18.9	0.0	24.3
Radix balthica	8.2	1.1	0.5	2.3	12.1

5.2 Aineiston luokittelu multinomiaalisella logistisella regressiolla

Neuroverkkojen lisäksi luokittelu tehdään multinomiaalisella logistisella regressiolla. Multinomiaalisen logistisen regressiomallin muodostamisessa käytetään kahta eri R-pakettia `VGAM` ja `nnet`, jota käytettiin myös yhden piilokerroksen neuroverkon estimointiin. `VGAM`-paketin `vglm`-funktion avulla voidaan estimoida mallin parametrit. Kyseistä funktiota käytetään yleisesti yleistettyjen lineaaristen mallien estimoinnissa. Funktio käyttää estimoinnissa SU-menetelmää ja Newtonin algoritmia. Multinomiaalinen logistinen regressiomalli estimoidaan myös `nnet`-paketin `multinom`-funktiota käyttäen, joka estimoi mallin parametrit käyttäen hyväksi `nnet`-funktiota. `nnet`-funktiota käytettiin myös yhden piilokerroksen neuroverkon estimoinnissa.

Kuten neuroverkoillakin luokitellessa, niin myös nyt luokittelu aloitetaan jakamalla aineisto opetus- ja testiaineistoksi. Opetusaineiston avulla estimoidaan mallin parametrit, ja estimoitua mallia käytetään testiaineiston luokitteluun. Luokittelijan hyvyyttä testataan samalla tavoin kuin neuroverkkojen tapauksessa, eli toistetaan luokittelu 100 kertaa ja lasketaan keskimääräinen luokitteluvirhe ja keskimääräinen sekaannusmatriisi.

Kuvassa 8 on `VGAM`-pakettia ja multinomiaalista logistista regressiota käyttäen saatujen luokitteluvirheiden jakauma, kun on tehty 100 toistoa. Luokitteluvirheiden keskiarvo on 0.146 ja keskihajonta 0.023. Jos verrataan tätä luokitteluvirhettä edellä laskettuihin neuroverkoilla saatuihin luokitteluvirheisiin, niin huomataan, että neuroverkoilla saadut tulokset ovat parempia. Usean piilokerroksen neuroverkkomallin luokitteluvirheiden keskiarvo on noin 0.054 yksikköä pienempi kuin multinomiaalisella logistisella regressiolla.



Kuva 8: Toistamalla saatujen luokitteluvirheiden jakauma multinomiaalisella logistisella regressiolla (`VGAM`-paketin `vglm`-funktio).

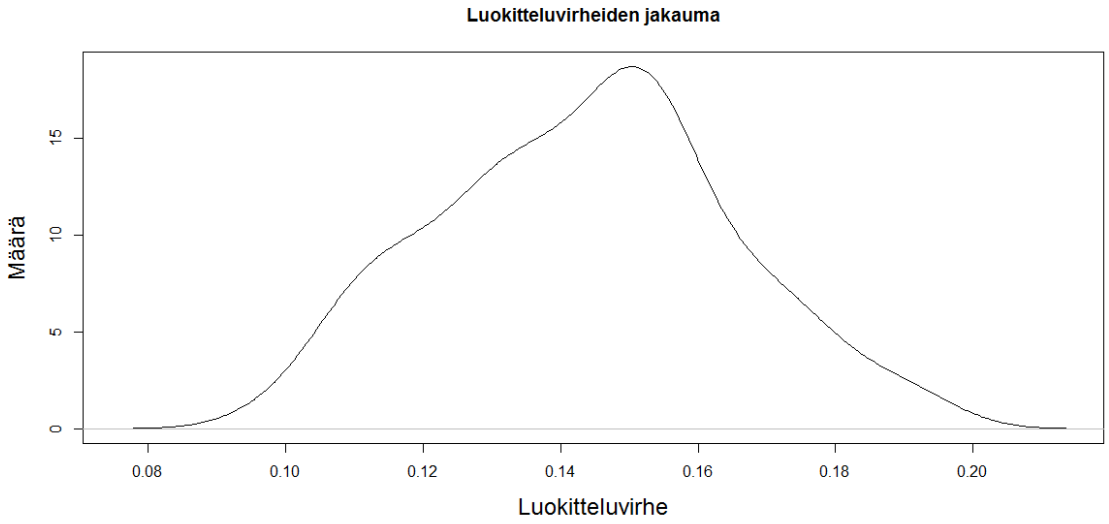
Taulukosta 6 nähdään, että myös multinomiaalinen logistinen regressiomalli tekee luokitellessa eniten virheitä samojen lajien kohdalla kuin neuroverkotkin. Lajikohtaiset oikein luokittelu prosentit ovat seuraavat: *Bithynia tentaculata* luokittuu oikein n. 89 %:n, *Myxas glutinosa* 88 %:n, *Physa acuta* 75 %:n ja *Radix balthica* 31 %:n tarkkuudella. Edelleen *Bithynia tentaculata* ja *Myxas glutinosa* luokittuvat parhaiten ja *Physa acuta* hieman huonommin kuin edellä mainitut. Huonoiten luokittuu edelleen *Radix balthica* mutta nyt hieman paremmin kuin usean piilokerroksen neuroverkolla. Usean piilokerroksen neuroverkolla kyseisestä lajista luokittui oikein alle 20 %.

Taulukko 6: Keskimääräinen sekaannusmatriisi multinomiaalisella logistisella regressiolla (`VGAM`-paketin `vglm`-funktio).

	Bithynia tentaculata	*Myxas glutinosa*	*Physa acuta*	*Radix balthica*	Yhteensä
Bithynia tentaculata	130.8	7.3	5.4	3.0	146.4
Myxas glutinosa	7.0	100.3	4.7	2.7	113.7
Physa acuta	3.3	2.1	18.6	0.7	24.7
Radix balthica	4.7	1.4	2.1	3.7	11.8

Seuraavaksi multinomiaalisella logistisella regressiolla luokittelu suoritetaan `nnet`-paketin `multinom`-funktiolla. Kuvassa 9 on nnet-paketin multinom-funktion avulla saadut luokitteluvirheet. Luokitteluvirheiden keskiarvo on 0.144 ja keskihajonta 0.021. Tulokset ovat hyvin samansuuntaiset kuin VGAM-paketin avulla saadut,

joten näyttäisi, että R-funktion valinnalla ei ole niin suurta merkitystä kuin itse menetelmän valinnalla. Myös nyt multinomiaalisella logistisella regressiolla tehty luokittelu onnistui heikommin kuin neuroverkoilla.



Kuva 9: Toistamalla saatujen luokitteluvirheiden jakauma multinomiaalisella logistisella regressiolla (`nnet`-paketin `multinom`-funktio).

Taulukosta 7 nähdään, että tämäkin luokittelumalli tekee eniten virheitä *Physa acutan* ja *Radix balthican* luokittelussa. Näyttää edelleen siltä, että mitä enemmän havaintoja lajista on, sitä paremmin malli lajin yksilöt luokittelee. Verrattaessa taulukossa 7 olevaa sekaannusmatriisia taulukossa 6 olevaan, huomataan, että ne ovat hyvin samankaltaiset. Tämäkin tulos puoltaa sitä, että R-funktion valinnalla ei näyttäisi olevan niin suurta merkitystä kuin menetelmän valinnalla.

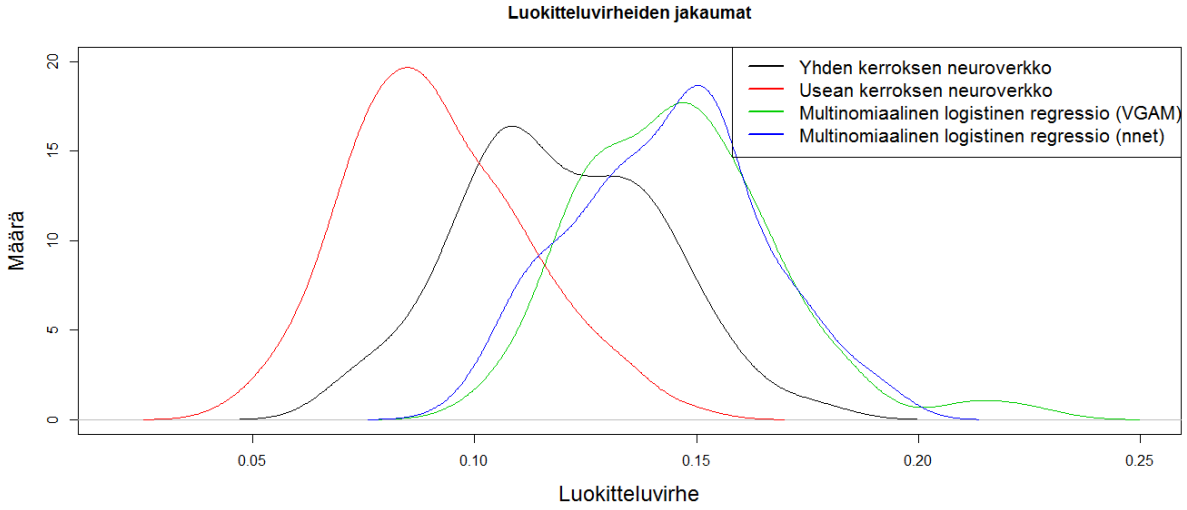
Taulukko 7: Keskimääräinen sekaannusmatriisi multinomiaalisella logistisella regressiolla (`nnet`-paketin `multinom`-funktio).

	Bithynia tentaculata	*Myxas glutinosa*	*Physa acuta*	*Radix balthica*	Yhteensä
Bithynia tentaculata	131.1	5.5	4.3	4.8	145.7
Myxas glutinosa	6.3	100.8	4.1	3.7	114.9
Physa acuta	3.0	1.7	18.5	1.3	24.5
Radix balthica	5.3	1.4	1.4	3.8	11.9

5.3 Parhaan luokittelumenetelmän etsiminen

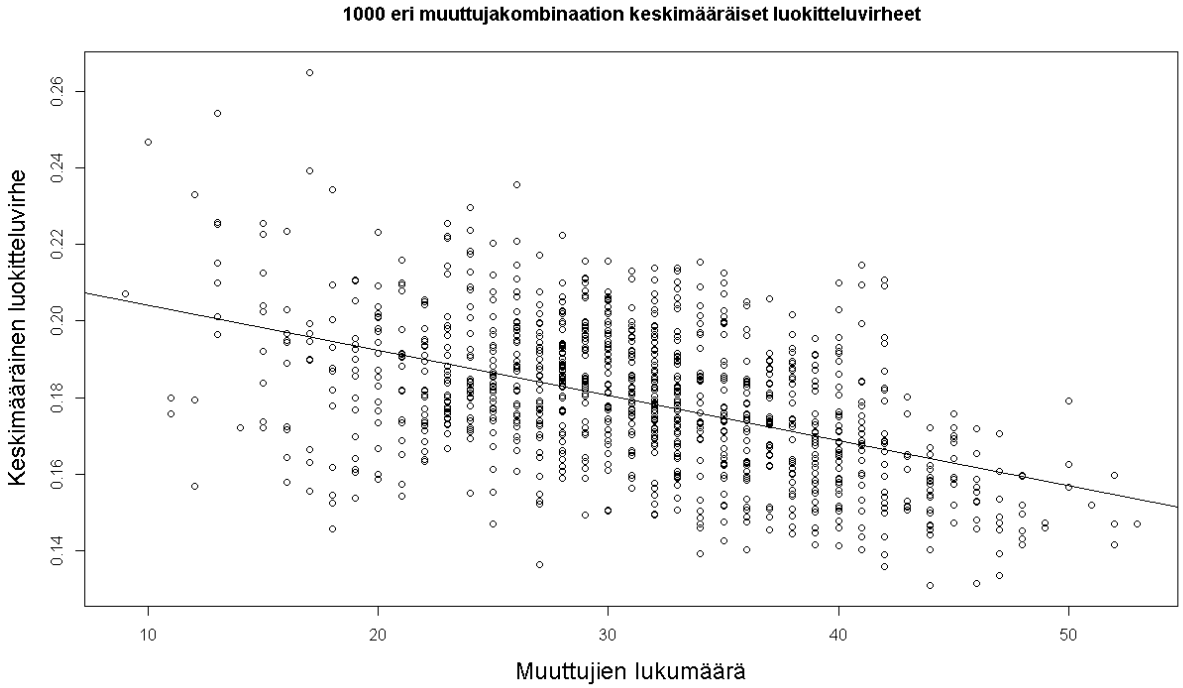
Paras luokittelumalli pyritään löytämään edellä mainittuja luokittelumalleja muokkaamalla. Tähän asti sovitetuista malleista usean piilokerroksen neuroverkko on toiminut kaikkein parhaiten. Parasta luokittelumallia etsittäessä tärkeimpänä tekijänä on optimaalisen selittävien muuttujien yhdistelmän valinta. Kaikissa tähän asti esitellyissä malleissa ovat olleet kaikki aineiston selittävät muuttujat mukana. Tällöin estimoitavia parametreja on ollut paljon suhteessa aineiston kokoon. Voisi olettaa, että vähentämällä selittäviä muuttujia päästään parempiin tuloksiin. Selittävien muuttujien valinnan lisäksi tutkitaan myös, miten eri funktioiden säätöparametrien valinta vaikuttaa luokittelumallin toimintaan.

Kuvasta 10 nähdään tähän asti esiteltyjen mallien luokitteluvirheiden jakaumat. Kuvasta voidaan havaita, että usean piilokerroksen neuroverkon luokitteluvirheiden jakauma on eniten vasemmalla, ts. sillä on keskimäärin saatu pienimpiä luokitteluvirheitä. Usean piilokerroksen neuroverkolla keskimääräinen luokitteluvirhe on hieman yli 0.09. Aluksi tutkitaan, päästäänkö tätä parempaan tulokseen yhden piilokerroksen neuroverkolla tai multinomiaalisella logistisella regressiolla jonkinlaisia muuttuja- ja parametrivalintoja käyttäen. Jos ei päästä, voidaan usean piilokerroksen neuroverkko todeta tässä tapauksessa parhaaksi luokittelumenetelmäksi.



Kuva 10: Eri menetelmillä toistoilla saatujen luokitteluvirheiden jakaumat.

Optimaalisen muuttujakombinaation löytämiseksi käytetään `R`-ohjelmiston `klaR`-paketin (Weihs et al., 2005) `stepclass`-funktiota ja SAS (SAS Institute Inc., 2003) `proc logistic` -proseduurista löytyvää muuttujavalintaa. Ennen varsinaista muuttujavalintaa tutkitaan, kuinka suuri merkitys erilaisilla muuttujavalinnoilla voi olla luokittelun tuloksiin. Tätä tutkitaan valitsemalla 1000 erilaista muuttujakombinaatiota satunnaisesti ja laskemalla jokaiselle selittävien muuttujien kombinaatioille keskimääräinen luokitteluvirhe.



Kuva 11: Keskimääräiset luokitteluvirheet eri muuttujamäärillä. Luokittelumallina on multinomiaalinen logistinen regressio. Kuvioon on sovitettu pns-suora.

Kuvasta 11 nähdään, että muuttujan valinnalla näyttäisi olevan merkitystä, koska hajontaa y-akselin suunnassa on paljon. Lisäksi kuvioon sovitetusta pns-suorasta voidaan todeta, että valittaessa selittävät muuttujat satunnaisesti, niin mitä enemmän muuttujia mallissa on, sitä paremmin se keskimäärin toimii.

Edellä tehty muuttujien satunnainen valinta ei ole kovinkaan järkevää, koska mahdollisia muuttujakombinaatioita on miljardeja. Tästä syystä varsinainen muuttujien valinta tehdään käyttämällä `R`:n `stepclass`-funktiota ja SAS:n `proc logistic` -proseduuria, joka sovittaa multinomiaalisen logistisen regressiomallin. SAS:n avulla saadaan selittävien muuttujien määrä karsittua seitsemään. Muuttujina ovat `Area`, `Round`, `Mean.gray`, `Min.blue`, `StdDev.red`, `Max.red` ja `Skew.red`. Tällä muuttujien yhdistelmällä saadaan luokitteluvirheitä jonkin verran pienemmäksi kaikilla muilla menetelmillä paitsi usean piilokerroksen neuroverkolla. Taulukosta 8 nähdään tarkemmin muuttujien valinnan vaikutus.

`Stepclass`-funktiota käytettäessä funktio antaa jokaisella kerralla hieman erilaisen muuttujien yhdistelmän. Tästä syystä muuttujien valinta tehdään useaan kertaan, ja jokaiselta kerralta lasketaan keskimääräinen luokitteluvirhe neuroverkoilla ja multinomiaalisella logistisella regressiolla. Sen jälkeen katsotaan, millä muuttujayhdistelmällä saadaan pienin luokitteluvirhe ja valitaan tämä muuttujayhdistelmä jatkoa varten. Muuttujanvalinta ja luokitteluvirheiden laskeminen tehdään useaan kertaan myös siitä syystä, että `stepclass`-funktiota käytettäessä ei suoranaisesti käytetä neuroverkkoja tai multinomiaalista logistista regressiota, vaan sii-

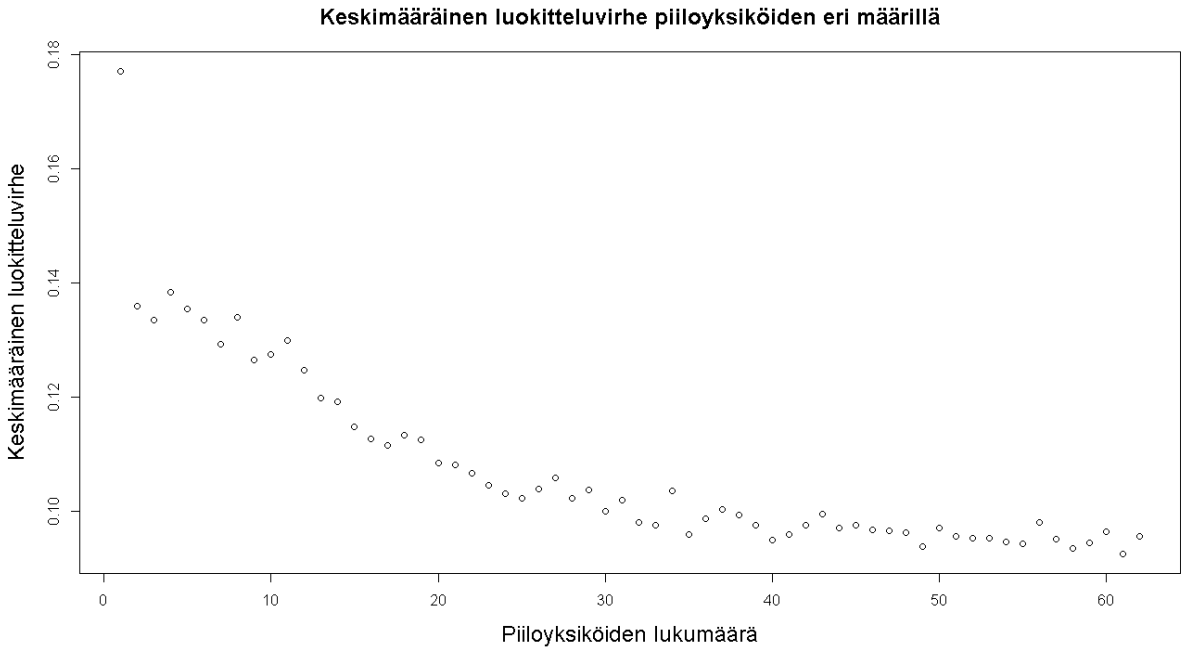
nä käytetään lda- tai qda-menetelmää luokittelijana (Hastie et al., 2009). Tästä syystä suoraan `stepclass`-funktiosta saadun tiedon perusteella ei voida päätellä, mikä muuttujien yhdistelmä toimii parhaiten neuroverkoilla tai multinomiaalisella logistisella regressiolla, vaan luokitteluvirheet täytyy erikseen laskea näillä menetelmillä. `Stepclass`-funktiota käyttämällä parhaimmaksi muuttujayhdistelmäksi saadaan: `Area`, `Minor`, `Min.gray`, `Min.blue`, `IntDen.blue`, `Median.blue`, `Mean.red`, `StdDev.red`, `Max.red`, `Median.green` ja `Skew.green`. Muuttujien määrä saadaan näin pudotettua yhteentoista.

Taulukosta 8 nähdään keskimääräiset luokitteluvirheet eri menetelmillä ja eri muuttujavalinnoilla. Multinomiaalinen logistinen regressio tehdään vain `nnet`-paketin avulla, koska tulokset `VGAM`-paketin kanssa sovitetun mallin kanssa näyttävät olevan samanlaisia, ja `nnet`-paketin funktiot ovat nopeampia kuin `VGAM`-paketin. Taulukosta nähdään, että multinomiaalisella logistisella regressiolla ei päästä yhtä hyviin tuloksiin kuin neuroverkoilla. Sekä yhden että usean piilokerroksen neuroverkoilla päästään parhaimmillaan n. 0.09 luokitteluvirheeseen.

Taulukko 8: Keskimääräiset luokitteluvirheet eri menetelmillä saaduilla selittävien muuttujien yhdistelmillä.

	Keskimääräinen luokitteluvirhe (Keskihajonta)		
	Kaikki muuttujat	SAS:lla saadut muuttujat	`Stepclass`:lla saadut muuttujat
Yhden piilokerroksen neuroverkko	0.119 (0.023)	0.111 (0.017)	0.094 (0.017)
Usean piilokerroksen neuroverkko	0.092 (0.020)	0.141 (0.021)	0.104 (0.017)
Multinomiaalinen logistien regressio	0.146 (0.023)	0.121 (0.017)	0.114 (0.021)

Huomattavaa on, että usean piilokerroksen neuroverkon luokittelutulos ei näyttäisi paranevan selittäviä muuttujia vähentämällä, vaan päinvastoin, se näyttäisi huononevan. Toinen huomionarvoinen asia on, että yhden piilokerroksen neuroverkko näyttää toimivan sitä paremmin, mitä enemmän piiloyksiköitä neuroverkossa on. Kun muuttujia on `stepclass`:lla saadut 11, voidaan piiloyksiköitä ottaa mukaan 62. Tätä on havainnollistettu kuvassa 12.



Kuva 12: Eri piiloyksiköiden lukumäärillä saadut keskimääräiset luokitteluvirheet.

Luokittelumallien lopullista paremmuutta selvitetään `t`-testin avulla. Aluksi testattiin, onko yhden piilokerroksen neuroverkolla saatu tulos merkitsevästi parempi kuin multinomiaalisella logistisella regressiolla saatu. Testin `t`-arvoksi saadaan $\texttt{t} = -7.6011$. Täten `p`-arvo on pienempi kuin 0.01. Samankaltainen tulos saadaan, kun tehdään testi usean piilokerroksen neuroverkon ja multinomiaalisen logistisen regression välillä. Näin ollen voidaan sanoa, että neuroverkoilla saadaan keskimäärin parempia tuloksia kuin multinomiaalisella logistisella regressiolla.

Taulukosta 8 nähdään, että yhden piilokerroksen neuroverkko toimii parhaiten, kun selittävien muuttujien määrää on vähennetty ja usean piilokerroksen neuroverkko toimii parhaiten, kun kaikki selittävät muuttujat ovat mallissa mukana. Näiden kahden mallin keskimääräiset luokitteluvirheet ovat melko saman suuruiset. Keskimääräisten luokitteluvirheiden erojen tilastollista merkitsevyyttä testataan `t`-testillä. Jotta testi olisi mahdollisimman tehokas, tehdään luokitteluvirheiden eron testaaminen siten, että suoritetaan luokittelu 1000 kertaa molemmilla malleilla samalla testi- ja opetusaineisto jaolla ja tehdään parittainen `t`-testi. Näin saatu `t`-testisuureen arvo $\texttt{t} = 0.12$, joten tämän perustella ei voida sanoa kummalla menetelmällä saadaan keskimäärin pienempiä luokitteluvirheitä, mikä oli odotettavissa jo taulukon 8 perusteella. Lisäksi näillä 1000 kerralla on tarkasteltu kummalla menetelmällä saadaan useammin toista menetelmää pienempi luokitteluvirhe. Usean piilokerroksen neuroverkon luokitteluvirhe oli pienempi 507 kertaa ja yhden piilokerroksen neuroverkon 493 kertaa.

Kun verrataan yhden ja usean piilokerroksen neuroverkkoja toisiinsa, on hyvä tarkastella myös näiden sekaannusmatriiseja. Usean piilokerroksen neuroverkon keski-

määräinen sekaannusmatriisi löytyy taulukosta 5 ja yhden piilokerroksen neuroverkon taulukosta 9. Näistä taulukoista nähdään, että usean piilokerroksen neuroverkko lajittelee paremmin lajit *Bithynia tentaculata*, *Myxas glutinosa* ja *Physa acuta*. Erot menetelmien välillä ovat kuitenkin melko pieniä. Sen sijaan lajin *Radix balthica* yhden piilokerroksen neuroverkko luokittelee huomattavasti paremmin. Yhden piilokerroksen neuroverkko luokittelee lajin keskimäärin oikein 48 %:n tarkkuudella. Usean piilokerroksen neuroverkko luokittelee kyseisen lajin vain 19 %:n tarkkuudella oikein (taulukko 5).

Taulukko 9: Yhden piilokerroksen neuroverkon keskimääräinen sekaannusmatriisi, kun selittävien muuttujien määrää on vähennetty.

	Bithynia tentaculata	*Myxas glutinosa*	*Physa acuta*	*Radix balthica*	Yhteensä
Bithynia tentaculata	138.4	3.1	3.1	2.1	146.7
Myxas glutinosa	4.6	106.3	1.0	1.5	113.4
Physa acuta	3.8	1.7	18.5	0.5	24.5
Radix balthica	3.7	1.8	0.9	6.0	12.4

6 Yhteenveto

Kotiloaineiston koneellinen luokittelu tehtiin käyttäen luokittelumallina neuroverkkoa ja multinomiaalista logistista regressiota. Molemmat menetelmät tehtiin kahdella eri `R`-funktiolla, jotta saatiin tietoa vaikuttaako luokittelun hyvyyteen tapa, jolla malli sovitetaan. Valitulla funktiolla ei näytä olevan merkitystä, mutta menetelmän valinnalla on. Neuroverkoilla aineisto saatiin luokiteltua yli 90 %:n tarkkuudella, kun multinomiaalisella logistisella regressiolla jäätiin hieman alle 90 %:n. Erot menetelmien välillä eivät olleet kovinkaan suuret, mutta tästä huolimatta voidaan sanoa, että neuroverkoilla päästiin keskimäärin parempiin tuloksiin. Myös Paliwalin ja Kumar toteavat artikkelissaan neuroverkkojen useammin olevan perinteisiä menetelmiä parempia kuin huonompia (Paliwal & Kumar, 2009).

Aineistossa on yhteensä 65 selittävää muuttujaa. Muuttujia on paljon suhteessa aineiston kokoon ja tästä syystä muuttujia on pyritty vähentämään käyttäen `stepclass` `R`-funktiota. Usean piilokerroksen neuroverkko toimii parhaiten, kun mukana on kaikki muuttujat, mutta muut luokittelumallit toimivat paremmin käytettäessä stepclass-funktion antamaa muuttujayhdistelmää. Funktion avulla saatu optimaalinen muuttujayhdistelmä on: `IntDen.blue`, `Min.gray`, `Min.blue`, `Area`, `Median.blue`, `Mean.red`, `StdDev.red`, `Max.red`, `Area`, `Median.green` ja `Skew.green`.

Aineistossa oli muutaman lajin osalta melko vähän havaintoja. Nämä lajit luokittuivat pääsääntöisesti huonommin. Luokittelutulosta saisi luultavasti parannettua, jos kerättäisiin lisää havaintoja lajeista *Physa acuta* ja *Radix balthica* ja sovitettaisiin luokittelumalli uudelleen. Esimerkiksi usean piilokerroksen neuroverkko luokitteli kotilot keskimäärin oikein n. 91 %:n tarkkuudella mutta lajin *Radix balthica* vain 19 %:n tarkkuudella.

Vaikka luokittelutulokset ovat melko hyviä, niin on hyvä muistaa, että tutkimusongelman ratkaisuun voi liittyä yleistettävyysongelma; vaikka malli toimii tutkittavassa aineistossa hyvin, se ei takaa sitä, että malli toimii yleisesti hyvin. Esimerkiksi ei voida olla täysin varmoja siitä, että tutkittavassa aineistossa hyvin toimiva malli toimisi hyvin joistain toisista vesistöistä poimitussa aineistossa. Tästä syystä tehtyjä luokittelumalleja olisi hyvä vielä testata jostain muusta vesistöstä eri ajankohtana poimitulla aineistolla. Näin saataisiin varmuutta siihen, että malli toimii myös yleisesti hyvin.

Kiitokset

Haluaisin kiittää tutkielman ohjaaja FT Salme Kärkkäistä asiantuntevasta avusta tutkielman teon aikana sekä professori Antti Penttistä tutkielmaani liittyvistä huomioista. Lisäksi kiitos aineiston tuottaneille lehtori Heikki Hämäläiselle, tutkija Timo Ruokoselle, tutkija Tuomas Turpeiselle ja vanhempi tutkija Kristian Meissnerille.

Kirjallisuutta

Abràmoff, D. M. D., Magalhães, D. P. J., & Ram, D. S. J. (2004). Image processing with ImageJ. *Biophotonics International*, 11(7):36–42.

Anderson, D. & McNeill, G. (1992). Artificial neural networks technology. *Kaman Sciences Corporation*, 258:17–19.

Cowan, J. D. (1967). A mathematical model of central nervous activity. PhD thesis, University of London.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. Springer, New York, 2. edition.

Hebb, D. O. (1949). *The Organization of Behavior*. Wiley, New York.

Hinton, G. E. & Sejnowski, T. J. (1986). Learning and relearning in Boltzmann machines. *Mit Press, Cambridge, Mass*, 1:282–317.

Hornik, K., Buchta, C., & Zeileis, A. (2009). Open-source machine learning: R meets Weka. *Computational Statistics*, 24(2):225–232.

Joutsijoki, H. & Juhola, M. (2012). Dagsvm vs. dagknn: An experimental case study with benthic macroinvertebrate dataset. In Perner, P., editor, *Machine Learning and Data Mining in Pattern Recognition*, volume 7376 of *Lecture Notes in Computer Science*, pages 439–453. Springer, Berlin.

Joutsijoki, H., Meissner, K., Gabbouj, M., Kiranyaz, S., Raitoharju, J., Ärje, J., Kärkkäinen, S., Tirronen, V., Turpeinen, T., & Juhola, M. (2014). Evaluating the performance of artificial neural networks for the classification of freshwater benthic macroinvertebrates. *Ecological Informatics*, 20:1–12.

McCulloch, W. S. & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133.

Mehrotra, K., Mohan, C., & Ranka, S. (1997). *Elements of Artificial Neural Networks*. Complex Adaptive Systems Series. Mit Press.

Minsky, M. & Papert, S. (1969). *Perceptrons*. Mit Press, Cambridge.

Paliwal, M. & Kumar, U. A. (2009). Neural networks and statistical techniques: A review of applications. *Expert Systems with Applications*, 36(1):2–17.

R Development Core Team (2008). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Rasband, W. S. (1997). ImageJ manual.
http://rsbweb.nih.gov/ij/docs/menus/analyze.html.

Retherford, R. D. & Choe, M. K. (1993). *Multinomial logit regression.* Wiley Online Library.

Ripley, B. D. (2007). *Pattern recognition and neural networks.* Cambridge University Press, Cambridge, UK.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408.

Rosenblatt, F. (1962). Principles of neurodynamics: perceptrons and the theory of brain mechanisms. Report (Cornell Aeronautical Laboratory). Spartan Books.

SAS Institute Inc. (2003). SAS/STAT Software, version 9.1. Cary, NC.

Sun, R. (1999). Artificial intelligence: Connectionist and symbolic approaches. University of Missouri-Columbia, Columbia.

Venables, W. N. & Ripley, B. D. (2002). *Modern Applied Statistics with S*, 4th edition. Springer, New York.

Weihs, C., Ligges, U., Luebke, K., & Raabe, N. (2005). klaR analyzing German business cycles. In Baier, D., Decker, R., & Schmidt-Thieme, L., editors, *Data Analysis and Decision Support*, pages 335–343, Berlin. Springer-Verlag.

Widrow, B. & Hoff, M. (1960). Adaptive switching circuits. *IRE WESCON Convention record*, 4:96–104. Reprinted in Andersen and Rosenfeld (1988).

Yee, T. W. (2013). *VGAM: Vector Generalized Linear and Additive Models.* R package version 0.9-3.

Ärje, J., Kärkkäinen, S., Turpeinen, T., & Meissner, K. (2013). Breaking the curse of dimensionality in quadratic discriminant analysis models with a novel variant of a Bayes classifier enhances automated taxa identification of freshwater macroinvertebrates. *Environmetrics*, 24(4):248–259.

Liitteet

Liite A: Aineiston muuttujat

RGB-piirteet	Harmaasävypiirteet	Geometriset piirteet
Mean.Blue	Mean.Grey	Area
StdDev.Blue	StdDev.Grey	X
Mode.Blue	Mode.Grey	Y
Min.Blue	Min.Grey	Perim
Max.Blue	Max.Grey	BX
XM.Blue	XM.Grey	BY
YM.Blue	YM.Grey	Width
IntDen.Blue	IntDen.Grey	Height
Median.Blue	Median.Grey	Major
Skew.Blue	Skew.Grey	Minor
Kurt.Blue	Kurt.Grey	Angle
Mean.Red		Circ
StdDev.Red		Feret
Mode.Red		Solidity
Min.Red		FeretX
Max.Red		FeretY
XM.Red		FeretAngle
YM.Red		MinFeret
IntDen.Red		AR
Median.Red		Round
Skew.Red		
Kurt.Red		
Mean.Green		
StdDev.Green		
Mode.Green		
Min.Green		
Max.Green		
XM.Green		
YM.Green		
IntDen.Green		
Median.Green		
Skew.Green		
Kurt.Green		

Liite B: R-koodi

```
#Aineiston lukeminen ja määrittelyt
data<-read.table("kotilot4.dat",header=T)
head(data)
names(data)
head(data)
data$Class<-as.factor(data$Class)

#Standardointi
for(i in 2:dim(data)[2]){
  data[,i]<-( data[,i]-mean(data[,i]) ) / sd(data[,i])
}
#Rapu-dummyn teko
rapu<-c( rep(0,63),rep(0,91),rep(1,119),rep(1,19),rep(1,10),rep(0,14),rep(0,46),
         rep(1,13),rep(0,43),rep(1,127),rep(0,25),rep(1,5),rep(1,19))
data<-cbind(data,rapu)
N<-nrow(data)
form<-as.formula(paste("Class~",paste(names(data[-1]), collapse="+")))

#############################################################################
#Kuvia muuttujista
names(data)
par(ask=T,cex.lab=1.5)
for(i in 2:(dim(data)[2]-2)){
pairs(data[i:(i+2)],col=data$Class,pch=as.numeric(data$Class))
}
par(mfrow=c(2,2),oma=c(0,0,1,0), cex.main = 2,cex.lab=1.8)
plot(data$Major,data$Minor,col=data$Class,pch=as.numeric(data$Class)
,xlab="Major",ylab="Minor")
plot(data$Height,data$Width,col=data$Class,pch=as.numeric(data$Class)
,xlab="Height",ylab="Width")
plot(data$Mean.gray.,data$Median.gray.,col=data$Class,
pch=as.numeric(data$Class),xlab="Mean.gray",ylab="Median.gray")
plot(data$Max.green.,data$Min.green.,col=data$Class,pch=as.numeric(data$Class)
,xlab="Max.green",ylab="Min.green")
title(main="Hajontakuvioita eri muuttujista",outer=T,cex=5)

#############################################################################
#Aineiston jakaminen opetus- ja testiaineistoksi

train.rows<-sample(1:N,0.5*N)
data.train<-data[train.rows,]
```

```
data.test<-data[-train.rows,]

#############################################################################
library(nnet)
#Testataan mikä on paras piiloyksiköiden lukumäärä
virheet<-NULL
piiloyksikot<-NULL
for(j in 1:100){
 train.rows<-sample(1:N,0.5*N)
 data.train<-data[train.rows,]
 data.test<-data[-train.rows,]

 for(i in 1:14){
  nn <- nnet(form,data = data.train,size=i,maxit=500)
  nn.pred<-predict(nn,newdata=data.test,type="class")
  Err<-1-sum(nn.pred==data.test$Class)/nrow(data.test)
  virheet[i]<-Err
  table(data.test$Class,nn.pred)
 }
 piiloyksikot[j]<-which.min(virheet)
}
table(piiloyksikot)

#Sovitetaan yhden piilokerroksen neuroverkkomalli ja testataan mallin hyvyyttä
nboot<-100
ok<-0
Err<-rep(100000,nboot)
confm<-NULL
#Asetetaan edistymisenseurantapalkki
pb <- txtProgressBar(min = 0, max = nboot, style = 3)

for(i in 1:nboot){
setTxtProgressBar(pb, i)
  while(ok==0){
  train.rows<-sample(1:N,0.5*N)
  data.train<-data[train.rows,]
  data.test<-data[-train.rows,]
  if(dim(table(data.train$Class))-dim(table(data.test$Class))==0) ok<-1
  }
  ok<-0
  #Valitaan piiloyksiköiden määräksi 14 edellä olevan tarkastelun perusteella
  nn <- nnet(form,data = data.train,size=14,maxit=500)
  nn.pred<-predict(nn,newdata=data.test,type="class")
```

```
  Err[i]<-1-sum(nn.pred==data.test$Class)/nrow(data.test)

  #Lasketaan keskimääräinen sekaannusmatriisi
  if(i==1) confm<-table(data.test$Class,nn.pred)
  if(i>1) confm<-confm+table(data.test$Class,nn.pred)
 }
close(pb)
#Keskimääräinen luokitteluvirhe
mean(Err)
sd(Err)
plot(density(Err),xlab="Luokitteluvirhe",ylab="Määrä",
main="Luokitteluvirheiden jakauma")

#Keskimääräinen sekaannusmatriisi
round(confm/nboot,1)
#Luokitteluvirhe
(sum(confm/nboot)-sum(diag(confm/nboot)))/sum(confm/nboot)

#Säilötään tulokset
nneterr<-Err

################################################################################
#Usean piilokerroksen neuroverkkoluokittelijan teko
library(RWeka)
MLP<-make_Weka_classifier("weka/classifiers/functions/MultilayerPerceptron")

#Luokitteluvirheen ja sekaannusmatriisin laskeminen toistoilla
ok<-0
nboot<-100
Err<-rep(100000,nboot)
pb <- txtProgressBar(min = 0, max = nboot, style = 3)
for(i in 1:nboot){
setTxtProgressBar(pb, i)
  while(ok==0){
  train.rows<-sample(1:N,0.5*N)
  data.train<-data[train.rows,]
  data.test<-data[-train.rows,]
  if(dim(table(data.train$Class))-dim(table(data.test$Class))==0) ok<-1
  }
  ok<-0
  NV<-MLP(form,data=data.train,control=Weka_control(V=20))
  NV.pred<-predict(NV,newdata=data.test)
  Err[i]<-1-sum(NV.pred==data.test$Class)/nrow(data.test)
```

```
  if(i==1) confm<-table(data.test$Class,NV.pred)
  if(i>1) confm<-confm+table(data.test$Class,NV.pred)
}
close(pb)
#Keskimääräinen luokitteluvirhe
mean(Err)
sd(Err)
plot(density(Err),cex.lab=1.5,,xlab="Luokitteluvirhe",ylab="Määrä",
main="Luokitteluvirheiden jakauma")

#Keskimääräinen sekaannusmatriisi
round(confm/nboot,1)
#Luokitteluvirhe
(sum(confm/nboot)-sum(diag(confm/nboot)))/sum(confm/nboot)

multineterr<-Err

#############################################################################
#Multinomiaalinen logistinen regressio. Mallin sovitus SU-menetelmällä.
library(VGAM)
confm<-NULL
ok<-0
nboot<-100
err<-rep(100000,nboot)
pb <- txtProgressBar(min = 0, max = nboot, style = 3)
for(j in 1:nboot){
setTxtProgressBar(pb, j)
  while(ok==0){
   train.rows<-sample(1:N,0.5*N)
   data.train<-data[train.rows,]
   data.test<-data[-train.rows,]
   if(dim(table(data.train$Class))-dim(table(data.test$Class))==0) ok<-1
  }
  ok<-0

  #Mallin sovitus
  multinom <- vglm(form,multinomial,data=data.train)
  summary(multinom)

  #Ennustetaan testiaineiston luokat
  multinom.pred<-predict(multinom,newdata=data.test)
  multinom.pred<-exp(multinom.pred)
```

```
  #Lasketaan mallin antamat todennäköisyydet kuulua tiettyyn luokkaan
  sum<-(1+multinom.pred[,1]+multinom.pred[,2]+multinom.pred[,3])
  bithten=round(multinom.pred[,1]/sum,2)
  myxasglu=round(multinom.pred[,2]/sum,2)
  physa=round(multinom.pred[,3]/sum,2)
  radixbal=round(1/sum,2)

  #ppp:ssä todenäköisyydet kuulua tiettyyn luokkaan
  ppp<-data.frame(bithten,myxasglu,physa,radixbal,data.test$Class)
  luokat<-c("Bithten","Myxasglu","Physa","Radixbal")

  #Katsotaan mihin luokkaan on suurin todennäköisyys kuulua
  multinom.pred.class<-NULL
  for(i in 1:dim(ppp)[1]){
  multinom.pred.class[i]<-luokat[which.max(ppp[i,1:4])]
  }

  #Luokitteluvirhe
  err[j]<-1-sum(multinom.pred.class==data.test$Class)/nrow(data.test)

  #Sekaannusmatriisi
  if(j==1) confm<-table(data.test$Class,multinom.pred.class)
  if(j>1) confm<-confm+table(data.test$Class,multinom.pred.class)
}
close(pb)
#Keskimääräinen luokitteluvirhe
mean(err)
sd(err)
plot(density(err),cex.lab=1.5,,xlab="Luokitteluvirhe",ylab="Määrä",
main="Luokitteluvirheiden jakauma")
#Keskimääräinen sekaannusmatriisi
round(confm/nboot,1)
(sum(confm)-sum(diag(confm)))/sum(confm)

multinomerrvgam<-err

#############################################################################
#Multinomaalinen logistinen regressio. Mallin sovitus käyttäen hyväksi
#neuroverkkoja.
library(nnet)

#Toistoilla saatu luokitteluvirhe
confm<-NULL
```

```
nboot<-100
Err<-rep(100000,nboot)
ok<-0
pb <- txtProgressBar(min = 0, max = nboot, style = 3)
for(i in 1:nboot){
setTxtProgressBar(pb, i)
  while(ok==0){
   train.rows<-sample(1:N,0.5*N)
   data.train<-data[train.rows,]
   data.test<-data[-train.rows,]
   if(dim(table(data.train$Class))-dim(table(data.test$Class))==0) ok<-1
  }
  ok<-0

#Sovitetaan multinomiaalinen logistinen regressiomalli ja
#testataan mallin hyvyyttä
multinom <- multinom(form,data=data.train,model=T)
multinom.pred<-predict(multinom,newdata=data.test)
Err[i]<-1-sum(multinom.pred==data.test$Class)/nrow(data.test)
if(i==1) confm<-table(data.test$Class,multinom.pred)
if(i>1) confm<-confm+table(data.test$Class,multinom.pred)

}
close(pb)

mean(Err)
sd(Err)
plot(density(Err),cex.lab=1.5,,xlab="Luokitteluvirhe",ylab="Määrä",
main="Luokitteluvirheiden jakauma")
round(confm/nboot,1)
(sum(confm)-sum(diag(confm)))/sum(confm)
multinomerrnnet<-Err

#################################################################
#Kaikkien menetelmien luokitteluvirheet samassa kuvassa

plot(density(nneterr),ylim=c(0,20),xlim=c(0.02,0.25),cex.lab=1.5,
xlab="Luokitteluvirhe",ylab="Määrä",main="Luokitteluvirheiden jakaumat")
lines(density(multineterr),col=2,cex.lab=1.5,
xlab="Luokitteluvirhe",ylab="Määrä",main="Luokitteluvirheiden jakauma")
lines(density(multinomerrvgam),col=3,cex.lab=1.5,,xlab="Luokitteluvirhe",
ylab="Määrä",main="Luokitteluvirheiden jakauma")
lines(density(multinomerrnnet),col=4,cex.lab=1.5,,xlab="Luokitteluvirhe",
```

```
ylab="Määrä",main="Luokitteluvirheiden jakauma")

legend(0.158,21,
c("Yhden kerroksen neuroverkko","Usean kerroksen neuroverkko",
"Multinomiaalinen logistinen regressio (VGAM)",
"Multinomiaalinen logistinen regressio (nnet)"),
lty=c(1,1),lwd=c(2.5,2.5,2.5,2.5),col=c(1,2,3,4),cex=1.3)

################################################################
#Optimaalisen muuttujayhdistelmän valinta
################################################################
#Etsitään parasta muuttujakombinaatiota stepclass-funktion avulla
library(klaR)

#Suoritetaan muuttujavalinta stepclass:n avulla
#Hyvyyden mittarina "correctness rate" eli kuinka suuren osan luokittelija
#luokittelee oikein. Luokittelumenetelmänä kokeiltu lda:ta ja qda:ta

#Koska eri kerroilta tulee erilaiset muuttujayhdistelmät, tutkitaan
#silmukan avulla kuinka erilaisia tuloksia niillä saadaan.

library(nnet)
reps<-20
virheet<-matrix(nrow=reps,ncol=2)
pb <- txtProgressBar(min = 0, max = reps, style = 3)
for(j in 1:reps){
setTxtProgressBar(pb, j)
form<-as.formula(paste("Class~",paste(names(data[-1]), collapse="+")))
stepc<-stepclass(form, data = data,direction = "both",maxvar=66,
criterion = "CR", fold = 10, method = "lda",improvement=0.0000000001)
form<-as.formula(stepc$formula)

nboot<-100
Err<-rep(100000,nboot)
ok<-0
for(i in 1:nboot){
  while(ok==0){
   train.rows<-sample(1:N,0.5*N)
   data.train<-data[train.rows,]
   data.test<-data[-train.rows,]
   if(dim(table(data.train$Class))-dim(table(data.test$Class))==0) ok<-1
  }
  ok<-0
```

```
#Sovitetaan multinomiaalinen logistinen regressio -malli ja
#testataan mallin hyvyyttä
multinom <- multinom(form,data=data.train,model=T)
multinom.pred<-predict(multinom,newdata=data.test)
Err[i]<-1-sum(multinom.pred==data.test$Class)/nrow(data.test)
}
virheet[j,1]<-mean(Err)
virheet[j,2]<-paste(form)[3]

}
close(pb)
virheet #Luokitteluvirheet eri muuttujayhdistelmillä

# Paras löydetty yhdistelmä stepclass:lla ("0.11037037037037")
form<-as.formula(Class ~ Area + Minor + Min.gray. + Min.blue. +
IntDen.blue. + Median.blue. + Mean.red. + StdDev.red. + Max.red.+
Median.green. + Skew.green.)

#SAS proc logistic -proseduurilla saadut muuttujat
form<-as.formula(Class~Area+Round+Mean.gray.+Min.blue.+StdDev.red.+
Max.red.+Skew.red.)

#########################################################################
#Tehdään luokittelu samalla testi/opetusaineisto jaolla eri
#menetelmillä ja katsotaan mikä on useimmiten paras.

nboot<-1000
ok<-0
Err1net<-rep(100000,nboot)
Erruseanet<-rep(100000,nboot)
Errlogit<-rep(100000,nboot)

pb <- txtProgressBar(min = 0, max = nboot, style = 3)
for(i in 1:nboot){
setTxtProgressBar(pb, i)
  while(ok==0){
  train.rows<-sample(1:N,0.5*N)
  data.train<-data[train.rows,]
  data.test<-data[-train.rows,]
  if(dim(table(data.train$Class))-dim(table(data.test$Class))==0) ok<-1
  }
  ok<-0
```

```
  #Yhden piilokerroksen neuroverkko
  form<-as.formula(Class ~ Area + Minor + Min.gray. + Min.blue. +
  IntDen.blue. + Median.blue. + Mean.red. + StdDev.red. +
  Max.red. + Median.green. + Skew.green.)
  nn <- nnet(form,data = data.train,size=62,maxit=500)
  nn.pred<-predict(nn,newdata=data.test,type="class")
  Err1net[i]<-1-sum(nn.pred==data.test$Class)/nrow(data.test)

  #Usean piilokerroksen neuroverkko
  form<-as.formula(paste("Class~",paste(names(data[-1]), collapse="+")))
  NV<-MLP(form,data=data.train,control=Weka_control(V=20))
  NV.pred<-predict(NV,newdata=data.test)
  Erruseanet[i]<-1-sum(NV.pred==data.test$Class)/nrow(data.test)


  #Multinomiaalinen logistinen regressio
  form<-as.formula(Class ~ Area + Minor + Min.gray. + Min.blue. +
  IntDen.blue. + Median.blue. + Mean.red. + StdDev.red. +
  Max.red. + Median.green. + Skew.green.)
  multinom <- multinom(form,data=data.train,model=T)
  multinom.pred<-predict(multinom,newdata=data.test)
  Errlogit[i]<-1-sum(multinom.pred==data.test$Class)/nrow(data.test)
 }
mean(Err1net)
mean(Erruseanet)
mean(Errlogit)
apu<-NULL
for(i in 1:nboot){
apu[i]<-which.min(c(Err1net[i],Erruseanet[i],Errlogit[i]))
}
table(apu)
t.test(Err1net,Errlogit,paired=T)
t.test(Err1net,Erruseanet,paired=T)
```

Liite C: Korrelaatiomatriiseja

Korrelaatiomatriisi harmaasävypiirteille. Riveillä ja sarakkeilla on samat muuttujat. Sarakkeiden nimistä on jätetty .gray pois.

	Mean	StdDev	Mode	Min	Max	XM	YM	IntDen	Median	Skew	Kurt
Mean.gray	1.00	-0.30	0.86	0.82	0.68	-0.46	-0.48	-0.30	0.99	-0.52	-0.21
StdDev.gray	-0.30	1.00	-0.34	-0.55	0.01	0.10	0.09	0.07	-0.32	-0.28	-0.64
Mode.gray	0.86	-0.34	1.00	0.66	0.54	-0.45	-0.45	-0.29	0.89	-0.53	-0.11
Min.gray	0.82	-0.55	0.66	1.00	0.53	-0.40	-0.42	-0.29	0.78	-0.07	0.04
Max.gray	0.68	0.01	0.54	0.53	1.00	-0.23	-0.27	-0.14	0.65	-0.33	-0.20
XM.gray	-0.46	0.10	-0.45	-0.40	-0.23	1.00	0.91	0.87	-0.46	0.08	-0.04
YM.gray	-0.48	0.09	-0.45	-0.42	-0.27	0.91	1.00	0.93	-0.47	0.07	-0.04
IntDen.gray	-0.30	0.07	-0.29	-0.29	-0.14	0.87	0.93	1.00	-0.28	-0.06	-0.10
Median.gray	0.99	-0.32	0.89	0.78	0.65	-0.46	-0.47	-0.28	1.00	-0.58	-0.21
Skew.gray	-0.52	-0.28	-0.53	-0.07	-0.33	0.08	0.07	-0.06	-0.58	1.00	0.70
Kurt.gray	-0.21	-0.64	-0.11	0.04	-0.20	-0.04	-0.04	-0.10	-0.21	0.70	1.00

Korrelaatiomatriisi sinisävypiirteille. Riveillä ja sarakkeilla on samat muuttujat. Sarakkeiden nimistä on jätetty .blue pois.

	Mean	StdDev	Mode	Min	Max	XM	YM	IntDen	Median	Skew	Kurt
Mean.blue	1.00	0.02	0.92	0.87	0.34	-0.36	-0.37	-0.15	0.99	-0.71	-0.59
StdDev.blue	0.02	1.00	-0.13	-0.28	0.40	-0.33	-0.34	-0.29	-0.01	-0.37	-0.45
Mode.blue	0.92	-0.13	1.00	0.80	0.27	-0.32	-0.33	-0.13	0.93	-0.59	-0.45
Min.blue	0.87	-0.28	0.80	1.00	0.15	-0.26	-0.27	-0.11	0.84	-0.43	-0.38
Max.blue	0.34	0.40	0.27	0.15	1.00	-0.01	-0.02	0.04	0.31	-0.12	-0.04
XM.blue	-0.36	-0.33	-0.32	-0.26	-0.01	1.00	0.92	0.87	-0.36	0.31	0.39
YM.blue	-0.37	-0.34	-0.33	-0.27	-0.02	0.92	1.00	0.91	-0.37	0.30	0.38
IntDen.blue	-0.15	-0.29	-0.13	-0.11	0.04	0.87	0.91	1.00	-0.15	0.08	0.17
Median.blue	0.99	-0.01	0.93	0.84	0.31	-0.36	-0.37	-0.15	1.00	-0.71	-0.57
Skew.blue	-0.71	-0.37	-0.59	-0.43	-0.12	0.31	0.30	0.08	-0.71	1.00	0.94
Kurt.blue	-0.59	-0.45	-0.45	-0.38	-0.04	0.39	0.38	0.17	-0.57	0.94	1.00

Korrelaatiomatriisi punasävypiirteille. Riveillä ja sarakkeilla on samat muuttujat. Sarakkeiden nimistä on jätetty .red pois.

	Mean	StdDev	Mode	Min	Max	XM	YM	IntDen	Median	Skew	Kurt
Mean.red	1.00	-0.44	0.80	0.76	0.52	-0.36	-0.37	-0.15	0.98	-0.46	0.11
StdDev.red	-0.44	1.00	-0.38	-0.70	0.06	0.26	0.25	0.17	-0.44	-0.09	-0.60
Mode.red	0.80	-0.38	1.00	0.54	0.36	-0.28	-0.27	-0.06	0.85	-0.56	0.06
Min.red	0.76	-0.70	0.54	1.00	0.30	-0.38	-0.38	-0.23	0.71	0.06	0.27
Max.red	0.52	0.06	0.36	0.30	1.00	-0.01	-0.00	0.06	0.48	-0.18	0.01
XM.red	-0.36	0.26	-0.28	-0.38	-0.01	1.00	0.92	0.87	-0.34	0.00	-0.23
YM.red	-0.37	0.25	-0.27	-0.38	-0.00	0.92	1.00	0.91	-0.34	-0.01	-0.22
IntDen.red	-0.15	0.17	-0.06	-0.23	0.06	0.87	0.91	1.00	-0.12	-0.12	-0.21
Median.red	0.98	-0.44	0.85	0.71	0.48	-0.34	-0.34	-0.12	1.00	-0.57	0.08
Skew.red	-0.46	-0.09	-0.56	0.06	-0.18	0.00	-0.01	-0.12	-0.57	1.00	0.37
Kurt.red	0.11	-0.60	0.06	0.27	0.01	-0.23	-0.22	-0.21	0.08	0.37	1.00

Korrelaatiomatriisi vihreäsävypiirteille. Riveillä ja sarakkeilla on samat muuttujat. Sarakkeiden nimistä on jätetty .green pois.

	Mean	StdDev	Mode	Min	Max	XM	YM	IntDen	Median	Skew	Kurt
Mean.green	1.00	-0.35	0.87	0.83	0.57	-0.39	-0.40	-0.21	0.99	-0.59	-0.21
StdDev.green	-0.35	1.00	-0.38	-0.60	0.07	0.10	0.10	0.06	-0.36	-0.11	-0.47
Mode.green	0.87	-0.38	1.00	0.68	0.44	-0.38	-0.39	-0.22	0.90	-0.60	-0.14
Min.green	0.83	-0.60	0.68	1.00	0.41	-0.35	-0.37	-0.23	0.79	-0.19	-0.01
Max.green	0.57	0.07	0.44	0.41	1.00	-0.13	-0.15	-0.08	0.54	-0.22	-0.03
XM.green	-0.39	0.10	-0.38	-0.35	-0.13	1.00	0.92	0.88	-0.38	0.09	-0.04
YM.green	-0.40	0.10	-0.39	-0.37	-0.15	0.92	1.00	0.92	-0.39	0.07	-0.05
IntDen.green	-0.21	0.06	-0.22	-0.23	-0.08	0.88	0.92	1.00	-0.21	-0.06	-0.12
Median.green	0.99	-0.36	0.90	0.79	0.54	-0.38	-0.39	-0.21	1.00	-0.65	-0.23
Skew.green	-0.59	-0.11	-0.60	-0.19	-0.22	0.09	0.07	-0.06	-0.65	1.00	0.66
Kurt.green	-0.21	-0.47	-0.14	-0.01	-0.03	-0.04	-0.05	-0.12	-0.23	0.66	1.00

Korrelaatiomatriisi geometrisille piirteille.

	Area	X	Y	Perim	BX	BY	Width	Height	Major	Minor
Area	1.00	0.92	0.96	0.95	0.91	0.91	0.93	0.93	0.98	0.95
X	0.92	1.00	0.92	0.93	0.99	0.99	0.99	0.81	0.94	0.93
Y	0.96	0.92	1.00	0.97	0.92	0.92	0.92	0.97	0.98	0.97
Perim	0.95	0.93	0.97	1.00	0.92	0.92	0.93	0.93	0.96	0.95
BX	0.91	0.99	0.92	0.92	1.00	0.99	0.99	0.81	0.94	0.93
BY	0.91	0.99	0.92	0.92	0.99	1.00	0.99	0.81	0.93	0.93
Width	0.93	0.99	0.92	0.93	0.99	0.99	1.00	0.82	0.95	0.93
Height	0.93	0.81	0.97	0.93	0.81	0.81	0.82	1.00	0.95	0.93
Major	0.98	0.94	0.98	0.96	0.94	0.93	0.95	0.95	1.00	0.96
Minor	0.95	0.93	0.97	0.95	0.93	0.93	0.93	0.93	0.96	1.00
Angle	0.04	0.03	0.02	0.02	0.02	0.02	0.03	0.02	0.03	0.04
Circ	-0.06	-0.10	-0.11	-0.25	-0.10	-0.10	-0.10	-0.10	-0.07	-0.03
Feret	0.97	0.94	0.99	0.97	0.94	0.93	0.95	0.95	1.00	0.96
FeretX	0.84	0.76	0.89	0.85	0.77	0.76	0.76	0.92	0.86	0.86
FeretY	0.64	0.57	0.66	0.65	0.58	0.58	0.58	0.66	0.64	0.64
FeretAngle	0.03	0.04	0.02	0.02	0.03	0.03	0.03	0.02	0.04	0.03
MinFeret	0.96	0.93	0.98	0.96	0.94	0.93	0.94	0.94	0.97	1.00
AR	0.52	0.50	0.53	0.51	0.48	0.48	0.50	0.51	0.59	0.38
Round	-0.47	-0.45	-0.47	-0.46	-0.43	-0.43	-0.45	-0.46	-0.54	-0.33
Solidity	0.17	0.12	0.13	0.00	0.13	0.13	0.13	0.12	0.17	0.22

	Angle	Circ	Feret	FeretX	FeretY	FeretA.	MinFer.	AR	Round	Solidity
Area	0.04	-0.06	0.97	0.84	0.64	0.03	0.96	0.52	-0.47	0.17
X	0.03	-0.10	0.94	0.76	0.57	0.04	0.93	0.50	-0.45	0.12
Y	0.02	-0.11	0.99	0.89	0.66	0.02	0.98	0.53	-0.47	0.13
Perim	0.02	-0.25	0.97	0.85	0.65	0.02	0.96	0.51	-0.46	0.00
BX	0.02	-0.10	0.94	0.77	0.58	0.03	0.94	0.48	-0.43	0.13
BY	0.02	-0.10	0.93	0.76	0.58	0.03	0.93	0.48	-0.43	0.13
Width	0.03	-0.10	0.95	0.76	0.58	0.03	0.94	0.50	-0.45	0.13
Height	0.02	-0.10	0.95	0.92	0.66	0.02	0.94	0.51	-0.46	0.12
Major	0.03	-0.07	1.00	0.86	0.64	0.04	0.97	0.59	-0.54	0.17
Minor	0.04	-0.03	0.96	0.86	0.64	0.03	1.00	0.38	-0.33	0.22
Angle	1.00	0.01	0.03	0.03	-0.33	0.67	0.03	-0.01	0.01	0.04
Circ	0.01	1.00	-0.12	-0.08	-0.06	-0.01	-0.07	-0.19	0.17	0.84
Feret	0.03	-0.12	1.00	0.86	0.65	0.03	0.97	0.60	-0.54	0.12
FeretX	0.03	-0.08	0.86	1.00	0.56	0.04	0.86	0.43	-0.39	0.15
FeretY	-0.33	-0.06	0.65	0.56	1.00	-0.37	0.64	0.33	-0.30	0.08
FeretAngle	0.67	-0.01	0.03	0.04	-0.37	1.00	0.03	0.03	-0.04	0.02
MinFeret	0.03	-0.07	0.97	0.86	0.64	0.03	1.00	0.41	-0.36	0.17
AR	-0.01	-0.19	0.60	0.43	0.33	0.03	0.41	1.00	-0.99	-0.09
Round	0.01	0.17	-0.54	-0.39	-0.30	-0.04	-0.36	-0.99	1.00	0.09
Solidity	0.04	0.84	0.12	0.15	0.08	0.02	0.17	-0.09	0.09	1.00