Tanja Juuti

Bayesiläiset graafiset mallit konenäössä: kukkivien kasvien lajintunnistus

Informaatioteknologian pro gradu -tutkielma

23. marraskuuta 2018

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Tanja Juuti

Yhteystiedot: tanja.i.juuti@student.jyu.fi

Ohjaajat: Matti Eskelinen ja Paavo Nieminen

Työn nimi: Bayesiläiset graafiset mallit konenäössä: kukkivien kasvien lajintunnistus

Title in English: Bayesian graphical models in computer vision: recognition of flowering plant species

Työ: Pro gradu -tutkielma

Suuntautumisvaihtoehto: Ohjelmistotekniikka

Sivumäärä: 109+11

Tiivistelmä: Suomen ympäristöhallinnossa on alettu viime vuosina kiinnostua enenevässä määrin konenäön hyödyntämisestä lajintunnistuksessa. Esimerkiksi tienvarsien haittakasvilajeja on tunnistettu varsin hyvin tuloksin. Koneellinen, lajeja tunnistava luokittelija voidaan toteuttaa esimerkiksi Bayes-verkoilla. Ne ovat todennäköisyysmalleja, joilla voidaan mallintaa muuttujien välisiä riippuvuuksia. Rakenteeltaan erilaiset Bayes-luokittelijat pystyvät eri tavoin huomioimaan piirremuuttujien välisiä riippuvuuksia. Esimerkiksi naiivi Bayes-menetelmä olettaa piirremuuttujien riippuvan ainoastaan lajiluokasta. Tutkimuksen tavoitteena olikin selvittää, millainen Bayes-verkko olisi sopiva luokittelija kukkivien kasvien lajintunnistukseen. Tunnistuskohteeksi valittiin kukat, koska ne ovat monimuotoisuutensa ja erottuvuutensa vuoksi hyviä kohteita elinympäristössään kuvattujen kasvien konenäölliseen lajintunnistukseen. Kukkien väriä kuvattiin värihistogrammien tilastollisilla tunnusluvuilla ja muotoa kuvamomenteilla sekä skaalainvarianteilla pistepiirteillä. Lisäksi kukista määritettiin terälehtien sijaintia ja kokoa kuvaavia piirteitä. Luokittelutulosten perusteella kuvista lasketut piirteet kuvasivat hyvin tutkimuksessa mukana olleiden lajien välisiä eroja. Sen sijaan piirremuuttujien riippuvuuksia huomioivat luokittelumallit eivät tuottaneet naiivia mallia parempia tuloksia. Onkin osoitettu, että tietynlaisissa riippuvuustilanteissa naiivi luokittelija on riippuvuutta huomioivien mallien veroinen. Riittävän yksinkertaisena naiivi malli on myös hyvä yleismalli. Osa tässä tutkimuksessa testatuista malleista oli liian monimutkaisia

havaintojen määrään nähden. Jatkossa tulisikin kiinnittää erityishuomiota siihen, millainen piirremuuttujien diskretisointi ja mallin rakennusprosessi tuottavat riittävän yksinkertaisen mutta hyvän mallin.

Avainsanat: konenäkö, Bayes-verkko, naiivi Bayes-luokittelija, koneoppiminen, lajintunnistus, luokittelu

Abstract: In recent years Finland's environmental administration has shown growing interest in utilizing computer vision in species recognition. As an example, good results have been achieved in identifying invasive plant species growing on roadsides. One option to build a classifier to recognize plant species is to use Bayesian networks, which are probability models that are able to model variable interdependencies. Different kinds of Bayesian classifiers have varying abilities to take into account the dependencies between feature variables. For example, naive Bayesian classifier is based on the assumption that the feature variables only depend on the object class. The aim of this study was to find out what kind of Bayesian network would be ideal in identifying flowering plant species. The images of flowers were used because the diversity and distinctness of flowers make them good targets for computer vision purposes. The color of the flower was represented by statistics of the color histogram and the shape of the flower was described with image moments and local scale-invariant features. Additional variables were extracted to represent the size and position of the petals. The classification results showed that the chosen features described the differences between the studied species quite well. However, the models taking into account the interdependencies of the feature variables did not yield any better results compared to the naive Bayesian classifier. In fact, other studies have showed that under certain dependence conditions the naive classifier can have discriminative power equal to the models that take the interdependencies into account. Naive Bayes is also simple enough to function as a good general model. Some models tested in this study were too complicated considering the small number of samples in the dataset. In future, special attention should be paid to how the feature variables should be discretized and how the model should be built in order to achieve a sufficiently simple but powerful model.

Keywords: computer vision, machine vision, Bayesian network, naive Bayesian classifier, machine learning, species recognition, classification

Kuviot

Taulukot

Sisältö

1 Johdanto

Tämän pro gradu -työn tarkoituksena on tutkia Bayes-verkkojen toimivuutta kuvassa esiintyvän kohteen luokittelussa liittyen erityisesti kukkien perusteella tapahtuvaan kasvien lajintunnistukseen. Bayes-verkot ovat koneoppimisessa yleisesti käytettyjä menetelmiä, joilla voidaan mallintaa muuttujien välinen riippuvuuksien verkko ja laskea esimerkiksi sairastumisriskin todennäköisyyksiä tiettyjen olosuhteiden vallitessa (Charniak 1991). Kohteen luokan tunnistaminen puolestaan on konenäkökentällä edelleen päänvaivaa aiheuttava kysymys, joten tunnistukseen soveltuvien menetelmien kehittäminen on tarpeen. Erilaisia luokkia on tyypillisesti paljon, niiden sisäinen variaatio voi olla suuri ja luokkien väliset erot voivat puolestaan olla hyvin pieniä, minkä vuoksi luokkien tunnistaminen on haasteellinen ongelma (Seeland ym. 2017; Zhang ym. 2013). Kirjallisuushaun perusteella erityyppisillä Bayes-verkoilla tapahtuvaa kukkivien kasvien lajintunnistusta ei ole juurikaan tutkittu, jos lukuun ei oteta naiivia Bayes-luokittelijaa.

Suomen ympäristöhallinnon piirissä on parin viime vuoden aikana virinnyt kiinnostusta konenäön hyödyntämiseen. Suomen ympäristökeskuksessa on parhaillaan käynnissä hanke *Konenäkö ympäristötiedon tuotannossa*, jonka tarkoituksena on tutkia konenäön soveltuvuutta lajintunnistukseen ja sen hyödyntämistä maastoinventoinneissa. Hankkeen kotisivuilla[1] on perusteltu hanketta ympäristönäytteenoton ja maastoinventointien työvoimavaltaisuudella ja suurilla kustannuksilla. Kartoituksissa hyödynnetään jo nyt satelliitti- ja ilmakuvia; tulevaisuudessa drone-laitteet helpottavat entisestään kuvadatan keräystä, ja niiden hyödyntämistä riistalaskennoissa tutkitaankin parhaillaan[2]. Kartoitusdatan käsittelyn automatisointi puolestaan vähentää inhimillisiä virheitä ja vakioi datankäsittelyprosessia, jolloin tulokset ovat keskenään vertailukelpoisempia. SYKE-hankkeen kotisivuilla[1] todetaan: *"Eri osaamisalueet yhdistämällä voidaan tuottaa sekä alueellista karttatietoa, ympäristön laatutietoa, että pidemmälle jalostettuja tuotteita, jotka palvelevat mm. kohteiden hoitotarpeen määrittelyä tai ympäröivän maankäytön suunnittelua."* Ympäristökeskuksen hanketta sivuaa Liikenneviras-

1. http://www.syke.fi/hankkeet/envision
2. https://www.helsinki.fi/fi/uutiset/kestava-kehitys/dronet-apuna-vesilintulaskennoissa

ton jo päättynyt tutkimus, jossa selvitettiin konenäön soveltuvuutta tienvarsikasvillisuuden inventointiin (Melander ym. 2016). Haittakasvilajeja tunnistettiin jopa 96 %:n tarkkuudella. Pro gradu -tutkielman aihe on siten varsin ajankohtainen ja aiheen valintaa voidaan perustella täsmälleen samoilla perusteilla, joilla on perusteltu edelläkuvattuja hankkeita. Wäldchen ja Mäder (2017) jopa visioivat artikkelissaan, että mobiililaitteiden avulla lajeja voisivat tunnistaa myös muut kuin asiantuntijat, mikä toisi arvokasta tietoa lajien levinneisyydestä biodiversiteetin ollessa uhattuna.

Maastoinventoinneissa kasvi esiintyy osana omaa elinympäristöään. Monessa kasvien lajintunnistusta konenäön keinoin käsitelleessä tutkimuksessa tunnistus on kuitenkin tapahtunut lehden muotoon perustuen, mikä vaatii laboratoriotyyppisissä olosuhteissa otettua kuvaa lehdestä, jotta lehti erottuisi selkeästi taustasta (Wäldchen ja Mäder 2017). Luonnollisessa ympäristössään lehteä on hankala erottaa samanväristen taustaobjektien massasta. Kuvattaessa kasvi omassa elinympäristössään kukka onkin lehteä helpommin taustamassasta eroteltavissa oleva kohde. Lisäksi kukat ovat monimuotoisia niin väriltään, muodoltaan kuin tekstuuriltaankin, minkä vuoksi ne soveltuvat hyvin koneellisesti tapahtuvaan lajintunnistukseen (Seeland ym. 2017). Siksi tämän tutkimuksen aineiston muodostavat lajin luonnollisessa elinympäristössä kukista otetut kuvat. On kuitenkin hyvä huomioida, että monimuotoisuus voi olla suurta myös samaan lajiin kuuluvilla kukilla, ja toisaalta erot lähilajien välillä voivat olla hädintuskin havaittavia perustuen yksittäiseen ominaisuuteen, mikä lisää tunnistamisen haasteellisuutta (Seeland ym. 2017).

Tutkimuksen tavoitteena on mallintaa luonnollisessa kasvuympäristössään kuvattujen kasvilajikuvien sisältämä data Bayes-verkoilla siten, että uudessa kuvassa oleva kukka pystytään luokittelemaan oikeaan lajiluokkaan. Koska sataprosenttiseen tarkkuuteen on mahdotonta päästä, tehtävää lähestytään kokeilemalla aineistoon monimutkaisuudeltaan eritasoisia Bayes-verkkoja. Lähtöhypoteesina mutkikkaamman verkon voidaan olettaa tiettyyn rajaan asti antavan parempia tuloksia. Liian monimutkainen malli kuitenkin ylisovittuu aineistoon, eli antaa hyviä tuloksia testiaineistossa, mutta alisuoriutuu uudessa aineistossa. Mutkikkaassa mallissa myös parametrien estimointi ja päättely hankaloituvat.

Toisaalta kuvadatan esittämiseen Bayes-verkkona liittyy myös kysymys, miten verkko tulee rakentaa ja kohteen ulkonäkö kuvata. Miten kunkin lajin ominaisuudet esitetään kyseiselle

lajille tunnusomaisena Bayes-verkkona? Kukista tulisi löytää sellaisia tunnusomaisia rakenteita, joiden sijainnin ja ulkomuodon perusteella voidaan rakentaa lajit toisistaan erotteleva Bayes-verkko.

Pro gradu -työn tavoitteet tiivistyvät seuraaviin tutkimuskysymyksiin:

1. Miten Bayes-verkko tulee rakentaa ja millaisilla piirteillä kohteen ulkonäköä kannattaa kuvata?
2. Miten hyvin erilaiset Bayes-verkot soveltuvat kasvilajintunnistukseen?

Tutkimuksen resurssien puitteissa ei ole mahdollista empiirisesti verrata Bayes-verkkoja muihin konenäkötutkimuksessa käytettäviin luokittelumenetelmiin. Tällainen vertailu toteutetaan siksi kirjallisuuteen perustuen osana loppuyhteenvetoa.

Tutkimuksen metodologinen viitekehys pohjautuu design science -tutkimukseen, joka voidaan nähdä myös konstruktiivisena tutkimuksena. Design science -tutkimus on yleinen metodologia informaatiotutkimuksen tieteenalalla (Hevner ym. 2004). Sen tavoitteena on olemassaoleviin teorioihin perustuen kehittää uusi artefakti, joka vastaa johonkin kohdealueen käytännön ongelmaan (Hevner ym. 2004). Tässä tutkimuksessa artefakti koostuu paitsi lajintunnistukseen soveltuvasta Bayes-verkosta, myös verkon rakentamisen prosessista. Design science -tutkimukseen sisältyy olennaisena osana kehittämis- ja evaluointisykli, joka luonteeltaan iteratiivisena auttaa omalta osaltaan tutkijaa paremmin ymmärtämään tarkasteltavaa ongelmaa (Hevner ym. 2004).

Pro gradu -tutkielma rakentuu seuraavasti: Luvut 2 ja 3 kattavat tutkimuksen teoreettisen taustan. Luvussa 2 kuvataan kohteen luokitteluun tähtäävä tunnistuksen yleisprosessi, tarkastellaan piirteiden muodostamisen ja mallien rakentamisen yleisperiaatteita sekä selvitetään, millaisia piirteitä kukkivien kasvilajien tunnistuksessa on yleensä käytetty. Luvun lopussa kuvataan tarkemmin muutama tässä tutkimuksessa sovellettu menetelmä. Luvussa 3 esitetään yleisteoria Bayes-verkoista. Luku 4 kattaa tutkimuksen empiirisen osion. Siinä kuvataan tutkimuksessa käytetyt mallit piirrevektoreineen, mallien antamat luokittelutulokset ja pohditaan menetelmän soveltuvuutta testiaineistoon. Luku 5 sisältää tulosten yhteenvedon ja pohdinnan tutkimuksen onnistumisesta.

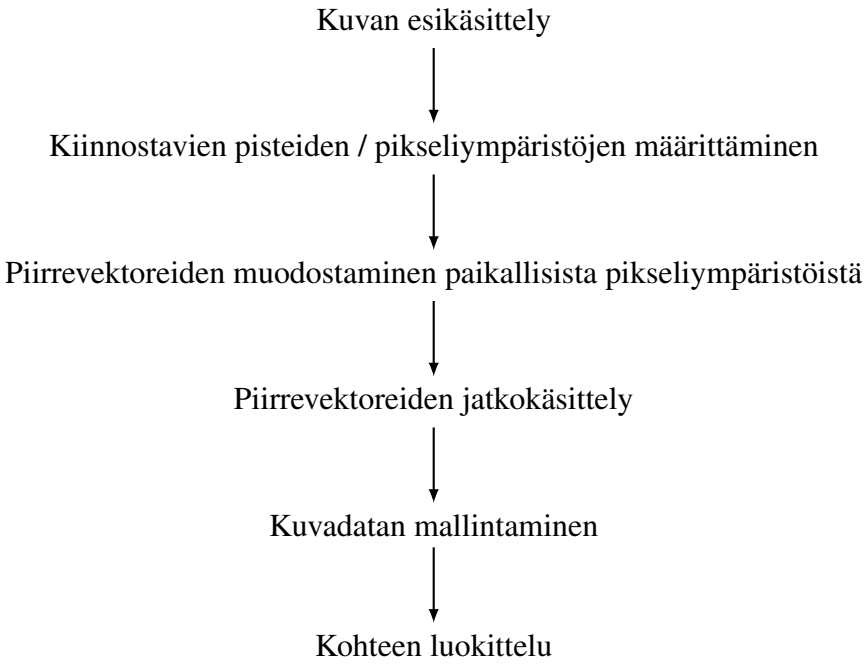
2 Kohteen tunnistamisen teoriaa ja menetelmiä

Szeliski (2011, luku 14) jakaa kohteen tunnistamisen kolmeen osa-alueeseen: kohteen tunnistaminen (engl. *object detection*), kohteen ilmentymän tunnistaminen (engl. *instance recognition*) ja kohteen luokan tunnistaminen (engl. *category-level object recognition*). Kohteen tunnistamisessa kuvasta pyritään etsimään ennaltamäärättyyn luokkaan kuuluvia kohteita, kuten kasvoja. Kohteen ilmentymän tunnistamisessa pyritään ratkaisemaan, löytyykö kuvasta aiemmin havaittu ilmentymä, mahdollisesti eri kuvakulmasta nähtynä. Kohteen luokan tunnistamisessa pyritään paitsi tunnistamaan kuvasta kiinnostavia kohteita, myös määrittämään löydetyille kohteille oikea luokka. Esimerkki kohteen luokan tunnistamisen sovellusalueesta on kasvin lajintunnistus kukasta otetun kuvan perusteella (Seeland ym. 2017).

2.1 Kohteen luokan tunnistusprosessin vaiheet

2.1.1 Prosessin yleiskuvaus

Pyrittäessä tunnistamaan kuvasta kohde ja määrittämään se oikeaan luokkaan yritetään itse asiassa kuvata matalan tason visuaalinen pikselidata merkityksellisinä korkean abstraktiotason käsitteinä (Zhang ym. 2013). Hyppyä datatasosta abstraktiotasolle kutsutaan semanttiseksi kuiluksi, jota kohteen tunnistusprosessin eri vaiheet kurovat umpeen askel askeleelta. Kohteen luokitteluprosessia ovat kuvanneet artikkeleissaan esimerkiksi Zhang ym. (2013) ja Seeland ym. (2017), joista jälkimmäinen keskittyy erityisesti kasvien lajintunnistusprosessiin kukkakuvien perusteella. Zhang ym.:n (2013) ja Seeland ym.:n (2017) kuvaama luokan tunnistusprosessin yleiskulku on esitetty mukailtuna kuviossa 1. Ennen kuin kuvasta lähdetään muodostamaan luokittelussa käytettäviä piirteitä, kuvaa on yleensä jo jollakin tapaa esikäsitelty, kuten muokattu väriavaruutta tai poistettu kohinaa; kuva voi olla myös segmentoitu. Piirteiden muodostaminen aloitetaan yleensä hyvin pienistä kuva-alueista, jotka muodostuvat kuvassa olevien kiinnostavien pisteiden ympärille. Paikallisilta alueilta lasketut piirteet jatkokäsitellään jollakin tapaa kuvaamaan isompia alueita kuvassa tai esimerkiksi tunnistettavan kohteen jotakin osaa. Isompia kuva-alueita tai kohteen osia voidaan edelleen koostaa yhteen, kunnes ollaan saavutettu haluttu, valittuun mallityyppiin sopiva yleistämis-



Kuvio 1: Kohteen luokan tunnistusprosessi mukailtuna Zhang ym.:n (2013) ja Seeland ym.:n (2017) artikkeleissaan kuvaamista prosesseista.

taso. Viimeisenä vaiheena on luokittelumallin rakentaminen. Malli opitaan koulutuskuvien avulla, joten oleellista on, että koulutuskuvissa esiintyvät kaikki kiinnostuksen kohteena olevat luokat riittävän monta kertaa. Lopulta koulutetulla mallilla voi vastata kysymykseen eli päätellä, mikä on uudessa kuvassa esiintyvän kohteen luokka.

2.1.2 Piirrevektoreiden muodostaminen

Koska prosessi aloitetaan tarkastelemalla matalan tason pikselidataa, ensimmäisenä askeleena on jollakin tavalla piirteistää sitä. Yksittäisten pikselien sijaan on yleensä hedelmällisempää piirteistää mielenkiintoisen pisteen ympärille muodostettuja pieniä laikkuja, joita kutsutaan pikseliympäristöiksi (Zhang ym. 2013). Mielenkiintoisia pisteitä ympäristöineen, kuten nurkkia, voidaan löytää useilla eri menetelmillä (Zhang ym. 2013). Eri menetelmät painottavat kuvadatan erilaisia ominaisuuksia, kuten intensiteettiä, värejä tai reunoja. Osa menetelmistä vain löytää mielenkiintoisia pisteitä, toiset menetelmät myös piirteistävät valmiiksi pikseliympäristön datan. Jälkimmäisiä menetelmiä kutsutaan paikallisten piirteiden deskriptoreiksi (engl. *local feature descriptors*).

Paikallisten piirteiden deskriptorien tulisi olla mahdollisimman invariantteja kohteesta riippumattomien tekijöiden suhteen (Zhang ym. 2013). Tällaisia tekijöitä ovat muun muassa valaistusolosuhteet, kuvakulma, skaalamuutokset, taustahäly ja kohteen osittainen peittyminen. Samaan aikaan piirteiden tulisi säilyttää mahdollisimman paljon kohteen ulkomuotoa kuvaavaa informaatiota, kuten dataa väristä, tekstuurista, muodosta ja suhteellisesta koosta. Zhang ym. (2013) erottelevat kohteet artikkelissaan (1) rakenteellisiin kohteisiin, jotka noudattavat yleensä tiettyä rakennetta ja muotoa jonkinasteisin variaatioin, ja (2) muodottomiin kohteisiin, joille ei voida määrittää täsmällistä muotoa. Esimerkkinä rakenteellisesta kohteesta on vaikkapa ihmisvartalo, jälkimmäisestä taivas. Kohteen muotoluokasta riippuu, minkätyyppistä informaatiota kohteen tunnistuksessa voidaan hyödyntää sekä minkälaiset kohteesta riippumattomat tekijät mahdollisesti haittaavat tunnistusta. Tässä tutkielmassa tunnistus keskittyy rakenteellisiin kohteisiin.

Koska yksittäinen kuva tuottaa yleensä suuren määrän paikallisia piirteitä, niitä harvemmin käytetään sellaisenaan. Tyypillisesti paikallisia piirrevektoreita koostetaan erilaisilla menetelmillä yhteen keskitason deskriptoreiksi (Zhang ym. 2013). Menetelmiä on useita, mutta yleensä ne koostuvat (1) muunnosvaiheesta, jossa paikalliset piirteet muokataan uudeksi dataksi, jonka katsotaan paremmin edustavan tutkittavaa ilmiötä — piirrevektoreista voidaan esimerkiksi muodostaa histogrammiesityksiä — sekä (2) spatiaalisesta yhdistämisvaiheesta, jossa uudet vektorit esimerkiksi keskiarvoistetaan kuvaamaan tiettyä kuva-alaa (Boureau ym. 2010). Tarvittava paikallisten piirteiden muokkausaste ja sopivan menetelmän valinta riippuvat muun muassa siitä, minkälaista mallia kohteen tunnistuksessa lopulta tullaan käyttämään. Esimerkiksi osapohjaisessa hierarkkisessa mallissa (ks. luku 2.2) paikalliset piirrevektorit on klusteroitu ulkomuoto- ja sijaintidatan perusteella, minkä jälkeen uuden datan muodostavat klusterikeskukset (Bouchard ja Triggs 2005). Piirrevektoreiden kuvaama kuva-ala voi niinikään olla kooltaan vaihteleva. Se voi kattaa koko kuvan tai osia siitä; osat taas voidaan muodostaa esimerkiksi yhdistämällä samankaltaisia pikseliympäristöjä toisiinsa tai osittamalla koko kuva säännölliseksi ruudukoksi (Zhang ym. 2013).

2.1.3 Mallin rakentaminen

Seuraavana tunnistusprosessin askeleena on sopivan mallin rakentaminen. Kohdetta voidaan tarkastella erilaisten ulkomuotomallien (engl. *appearance model*) pohjalta, ja kohteen ominaisuuksista riippuu, minkälainen ulkomuotomalli kannattaa valita (Zhang ym. 2013). Niin sanotut rakenteelliset ulkomuotomallit sopivat luokittelemaan rakenteellisia kohteita ja rakenteettomat mallit muodottomia kohteita (Zhang ym. 2013). Zhang ym. (2013) jakavat rakenteelliset mallit edelleen ikkunointimalleihin ja osapohjaisiin malleihin, joskin mainittuja malleja voidaan myös sekoittaa. Ikkunointimalleissa kohde rajataan ennaltamäärätyn muotoisella ikkunalla ja kohdetta kuvaavat piirteet lasketaan ikkunan sisältä. Osapohjaisessa mallissa kohde ositetaan pienempiin, kohteen rakenteen kannalta mielekkäisiin kokonaisuuksiin, joiden sisältä piirteet muodostetaan. Toisin kuin ikkunointimalli, osapohjainen malli sisältää informaatiota myös osien sijainnista toisiinsa nähden. Osapohjaisia malleja käsitellään tarkemmin luvussa 2.2.

Zhang ym. (2013) vertailevat artikkelissaan ikkuna- ja osapohjaisten mallien kykyä sopeutua luokansisäiseen vaihteluun tilanteessa, jossa kuvadatan oletetaan koostuvan invarianteista paikallisista piirteistä. Vertailussa ikkunapohjaiset mallit on jaettu globaalisti ja paikallisesti koostettuihin ikkunointimalleihin. Globaalien ikkunointimallien piirrevektorit kuvaavat koko ikkuna-alaa, joten ne kadottavat täysin spatiaalisen informaation, mutta kykenevät sen sijaan käsittelemään hyvin kohteen rotaatioita ja asentomuutoksia sekä kohtuullisesti kohteen osittaista peittymistä. Paikallisesti koostetut ikkunointimallit jakavat ikkunan osaruudukoihin ja koostavat piirrevektorit kussakin osaruudussa. Täten ne säilyttävät spatiaalista informaatiota, mutta huonona puolena ne eivät ole enää rotaatioinvariantteja ja kykenevät globaaleja ikkunointimalleja huonommin sopeutumaan myös kohteen osittaiseen peittymiseen. Osapohjaiset mallit sisältävät sekä globaalien että paikallisten ikkunointimallien hyviä ominaisuuksia. Mallintamalla osien sijainnin toisiinsa nähden ne sisällyttävät spatiaalisen informaation osajakoon, mitä voidaan hyödyntää tutkittaessa kohteen asentomuutoksia. Zhang ym. (2013) määrittelevätkin osapohjaiset mallit pikemminkin asentomuutokset tiedostaviksi kuin täysin invarianteiksi niille. Koska kohteen tunnistamiseen ei yleensä tarvita kaikkia osia, osapohjaiset mallit kykenevät kohtuullisen hyvin sietämään myös kohteen osittaisen peittymisen.

Sekä osapohjaiset että ikkunointimallit pystyvät Zhang ym.:n (2013) mukaan huomioimaan hyvin skaalamuutoksia sekä suhteellisen hyvin pieniä katselukulman muutoksia.

Koulutusprosessin viimeisessä vaiheessa rakennetaan luokittelumalli. Päämääränä on määrittää, miten mitatut piirre- eli ennustemuuttujat ovat yhteydessä ennustettavaan vastemuuttujaan eli tämän tutkimuksen tapauksessa lajiluokkaan, ja pystyä opitun eli estimoidun mallin avulla ennustamaan oikein uuden havainnon lajiluokka (Zaki ja Meira 2014). Käytännössä mallin estimointi tarkoittaa mallin rakenteen ja parametrien estimointia eli numeeristen arvojen määrittämistä parametreille tiettyjen mallin ominaisuuksien ollessa etukäteen määrättyjä (Myllymäki ja Tirri 1998). Etukäteen määrätään yleensä mallityyppi, kuten onko malli neuroverkko tai Bayes-verkko, ja mahdollisesti joitakin rajoitteita mallin rakenteelle (Myllymäki ja Tirri 1998). Esimerkki yksinkertaisesta mallista on lineaarinen, yhden ennuste- ja vastemuuttujan regressiomalli $y = \alpha + \beta x$ (Ranta, Rita ja Kouki 1997). Mallissa ennustavan muuttujan x ja vastemuuttujan y välille oletetaan jo etukäteen lineaarinen yhteys. Parametrien α ja β estimaatit määräävät lopullisesti, millainen lineaarinen yhteys muuttujien välillä täsmälleen on: onko suora loiva vai jyrkkä ja kulkeeko se origon kautta.

Mallin sopivuutta aineistoon mitataan mallinvalinta- eli sopivuuskriteerillä, ja yleensä lopulliseksi malliksi valitaan parhaan sopivuuskriteerin arvon tuottanut malli (Heckerman 1999; Myllymäki ja Tirri 1998). Sopivuuskriteereitä on erilaisia, mutta tyypillisesti niissä pyritään minimoimaan tai maksimoimaan funktion arvoa (Myllymäki ja Tirri 1998). Esimerkiksi edellä kuvatussa yksinkertaisessa regressiomallissa pyritään minimoimaan neliösumma, joka saadaan laskemalla vastemuuttujan ennustearvojen poikkemat havaituista arvoista (Ranta, Rita ja Kouki 1997). Yleensä sopivuuskriteeri sisältää myös ylisovittamisesta rankaisevan termin (Myllymäki ja Tirri 1998). Ylisovittaminen tarkoittaa, että mallin rakenne on liian monimutkainen sen kuvaamaan ilmiöön nähden: malli sopii tarkasti koulutusaineistoon mutta sen ennustekyky uusissa aineistoissa on huono — malli ei yleisty (Myllymäki ja Tirri 1998).

Luokittelussa käytettäviä mallityyppejä on useita ja niitä voidaan jaotella eri tavoin. Tämän tutkimuksen kannalta valaiseva on jako generoiviin (engl. *generative*) ja erotteleviin (engl. *discriminative*) luokittelijoihin (Xue 2008). Generoiviin luokittelijoihin kuuluvat esimerkiksi Bayes-verkot ja erotteleviin luokittelijoihin neuroverkot sekä regressiomalli, jonka vas-

temuuttuja on luokitteluasteikollinen (Rubinstein ja Hastie 1997). Generoivat luokittelijat mallintavat piirremuuttujien X ja luokkamuuttujan Y koko yhteistodennäköisyysjakauman $P(X,Y) \propto P(X|Y)P(Y)$ ja estimoivat malliparametrit sen perusteella. Malleja kutsutaan generoiviksi, koska niiden perusteella on mahdollista myös generoida uusia havaintoja luokista Y_i (Xue 2008). Erottelevat luokittelijat ovat kiinnostuneita ainoastaan luokkamuuttujan ehdollisesta todennäköisyydestä $P(Y|X)$ ehdolla havaittu data ja mallintavat parametriestimaatit siihen perustuen (Xue 2008). Siten erottelevat luokittelijat keskittyvät maksimoimaan suoraan luokkien välisiä eroja, kun taas generoivat luokittelijat pyrkivät mallintamaan kunkin luokan koko todennäköisyysjakauman (Rubinstein ja Hastie 1997). Tämän seurauksena erottelevissa luokittelijoissa kaikkia luokkia joudutaan käsittelemään samanaikaisesti, mikä tekee malleista hankalasti koulutettavia ja huonosti skaalautuvia (Rubinstein ja Hastie 1997). Generoiville luokittelijoille riittää sen sijaan käsitellä yhtä luokkaa kerrallaan, minkä vuoksi ne ovat suhteellisen helposti koulutettavia (Rubinstein ja Hastie 1997). Toisaalta generoivia luokittelijoita on kritisoitu siitä, että ne eivät keskity olennaiseen eli luokkien välisiin eroihin, minkä vuoksi on yleisesti ajateltu, että luokittelussa pitäisi suosia ensisijaisesti erottelevia luokittelijoita (Ng ja Jordan 2002). Suurilla havaintomäärillä erottelevat mallit tuottavatkin yleensä generoivia luokittelijoita parempia tuloksia (Ng ja Jordan 2002). Mutta muun muassa Ng ja Jordan (2002) havaitsivat, että vähäisellä koulutusdatan määrällä generoiva malli suoriutui luokittelusta erottelevaa mallia paremmin. Yhteenvetona voidaankin sanoa yhteistodennäköisyysjakauman mallintamisen tuovan lisää informaatiota, jos jakauma on oikea, mutta väärä jakauma altistaa luokitteluvirheille — vähemmän oletuksia sisältävinä erottelevat luokittelijat ovat siten generoivia luokittelijoita yleiskäyttöisempiä (Rubinstein ja Hastie 1997).

2.2 Katsaus osapohjaisiin malleihin

Osa on yleensä jokin suhteellisen kiinteä kokonaisuus kohteesta, kuten ihmisen käsivarsi tai pää. Sitä voidaan kuvailla erilaisilla geometrisilla ominaisuuksilla, kuten muodolla, tai esimerkiksi väreillä, tekstuurilla ja gradienteilla (Zhang ym. 2013). Osatopologia puolestaan kuvaa osien kiinnittymistä toisiinsa ja sijaintia suhteessa toisiin osiin (Zhang ym. 2013). Osapohjaisia malleja kuvataan tyypillisesti verkoilla, joissa solmut ovat osia ja solmujen vä-

liset kaaret kuvaavat osatopologian (Szeliski 2011, luku 14.4.2; Zhang ym. 2013). Verkkojen rakenteet vaihtelevat mutkikkuudeltaan. Tyypillisesti mutkikkaampi malli kykenee yksinkertaista paremmin kuvaamaan kohteen topologian, mutta samalla mallin oppiminen ja sen pohjalta tehtävä päättely hankaloituvat (Carneiro ja Lowe 2006).

Sekä Carneiro ja Lowe (2006) että Szeliski (2011, luku 14.4.2) esittelevät yksinkertaisimpana osapohjaisena mallina *bag of features* -mallin, joka ei lainkaan mallinna osien sijaintia toisiinsa nähden. Malli koostuu joukosta solmuja, joiden väliltä puuttuvat kaaret. Täysyhteydellisessä mallissa (engl. *constellation model*) kuvataan sen sijaan kaikki osien väliset parittaiset yhteydet (Szeliski 2011, luku 14.4.2). Sen ongelmana kuitenkin on, että parametrien määrä kasvaa eksponentiaalisesti suhteessa osien määrään, jolloin mallin oppimisesta tulee hankalaa jo vähäisellä määrällä osia (Carneiro ja Lowe 2006). Näiden kahden ääripään väliin sijoittuvat muun muassa tähtimallit, puumallit ja hierarkiamallit (esim. Szeliski 2011, luku 14.4.2). Puumallien (engl. *tree model*) kuvaamassa topologiassa ei-juurisolmujen sijainti riippuu vain solmun vanhempien sijainnista; ne soveltuvat hyvin kuvaamaan esimerkiksi ihmisten tai eläinten ruumiinrakennetta (Zhang ym. 2013). Tähtimalli (engl. *star model*) on puumallin erikoistapaus ja siinä on yksi keskussolmu, johon muut osat ovat kiinnittyneet (Fergus, Perona ja Zisserman 2005; Szeliski 2011, luku 14.4.2). Hierarkiamallissa (engl. *hierarchy model*) kohde jaetaan osiin, ja osat edelleen paikallisten piirteiden luokkiin; mallin alimman tason paikallisten piirteiden luokat vastaavat kuvista löydettyjä ilmiasultaan yhteneväisiä paikallisia klustereita (Bouchard ja Triggs 2005). Bouchard ja Triggs (2005) toteavat hierarkiamallin olevan edelleen tehokas, vaikka paikallisten piirteiden muodostamien osien määrä kohoaisi satoihin. Eräänlainen yleistys osapohjaisista malleista on kielioppipohjainen malli, joka mahdollistaa kohteen osarakenteen muuntelun tietyissä kielioppisääntöjen mukaisissa rajoissa (Zhu ja Mumford 2006; Zhang ym. 2013).

2.3 Kohteen luokittelussa käytettävistä piirteistä

Luokiteltavan kohteen ominaisuuksista riippuen kohdetta kannattaa kuvata sille soveltuvilla piirteillä. Käydään siksi seuraavaksi lyhyesti läpi kukkien lajintunnistuksessa hyödynnettyjä piirteitä, ja lopuksi kuvataan tarkemmin muutama tässä tutkimuksessa hyödynnetty menetelmä.

2.3.1 Katsaus kasvilajintunnistuksessa hyödynnettyihin piirteisiin

Kukkien perusteella tapahtuvassa kasvilajintunnistuksessa käytettäviä piirteitä ovat kuvanneet katsausartikkelissaan Wäldchen ja Mäder (2017). Tutkimuksissa kukkia on kuvailtu muotoa, väriä ja tekstuuria kuvaavilla piirteillä. Yksinkertaisimpia kukkatutkimuksissa käytettyjä muotopiirteitä ovat kohteen geometrisia ominaisuuksia kuvaavat mitat, kuten pinta-ala, pyöreys, reunakäyrän pituus ja muotosuhde (engl. *aspect ratio*). Mutkikkaammat muotoa kuvaavat piirteet voidaan jakaa kohteen reunakäyrää kuvaaviin piirteisiin ja kohteen tai sen osan muotoa kokonaisuudessaan kuvaaviin piirteisiin. Koska jälkimmäiset hyödyntävät dataa kohteen koko alueelta, ne eivät ole niin herkkiä kohteen osittaiselle peittymiselle tai usean kohteen osittaiselle yhteensulautumiselle kuin reunakäyrän piirteet. Sekä kohteen reunakäyrää että kokonaismuotoa voidaan kuvata useilla erilaisilla menetelmillä.

Kappaleen kokonaismuotoa kuvaavat Hun kuvamomentit ovat suhteellisen yksinkertaisia. Ne ovat invariantteja skaalaukselle, siirrolle ja rotaatiolle (Hu 1962). Koska ne lasketaan yleensä kynnystetystä kuvasta, niiden huonona puolena on herkkyys varsinaisen kappaleen ulkopuolisille havaintosegmenteille (Wäldchen ja Mäder 2017).

Toinen tapa kuvata kappaleen muotoa ovat paikallisten piirteiden deskriptorit, joista kukkatutkimuksissa yleisesti käytettyjä ovat harmaasävykuvista gradienttihistogrammeja eri tavoin muodostavat SIFT (*scale-invariant feature transform*), SURF (*speeded up robust features*) ja HOG (*histogram of oriented gradients*) (Wäldchen ja Mäder 2017). Nämä piirteet voidaan muodostaa kuvasta löytyneiden avainpisteiden ympäristöstä tai laskea koko kuva-alueelta tasavälein näytteistettyjen pikseleiden ympäriltä (Seeland ym. 2017). Viimeiseksi mainittua, niin sanottua tiheää näytteistystä on suosittu, koska luokittelutulosten on todettu paranevan paikallisten piirteiden määrän kasvaessa, joskin sen on myös todettu olevan herkkä skaalan ja kuvakulman muutoksille (Hietanen ym. 2016). Seeland ym. (2017) totesivat kuitenkin piirredeskriptoreja vertailevassa tutkimuksessaan, että on riittävää muodostaa piirteet vain löydettyjen avainpisteiden ympäriltä tiheän näytteistyksen sijaan. Yksi tutkimuksen testiaineistoista oli tässäkin tutkimuksessa hyödynnetty kukkakuvakokoelma. HOG-menetelmää lukuunottamatta avainpisteiden ympärille muodostetut piirteet antoivat tiheää näytteistystä parempia tuloksia eritoten kuvaominaisuuksiltaan heterogeenisissä kokoelmissa. Piirredeskriptoreista SIFT antoi paremman luokittelutuloksen kuin SURF- ja HOG-menetelmät, mut-

ta avainpisteiden tunnistusvaihe kannatti tällöin toteuttaa DoH-menetelmällä (*determinant of the Hessian*), jota yleensä käytetään SURF-menetelmän kanssa. Tulokset kuitenkin vaihtelivat jonkin verran aineistosta riippuen. Esimerkiksi tässä tutkimuksessa hyödynnetylle aineistolle parhaimman luokittelutuloksen antoi SURF-menetelmä.

Kuten luvussa 2.1 mainittiin, paikallisten deskriptorien piirrevektoreita koostetaan yleensä jollakin tapaa yhteen dimension pienentämiseksi. Tällaisia kukkatutkimuksissa usein käytettyjä menetelmiä ovat *bag of features*, joka kuvataan tarkemmin luvussa 2.3.5, ja *Locality Constrained Linear Coding* (LLC) (Seeland ym. 2017; Wäldchen ja Mäder 2017). Osa menetelmistä olettaa, että piirredeskriptori voi kuulua vain yhteen koosteluokkaan, mutta esimerkiksi LLC-menetelmä sallii, että piirredeskriptori voi kuulua M-kappaleeseen koosteluokkia (Seeland ym. 2017).

Kukkien reunakäyrää on kuvattu muun muassa laskemalla etäisyys kappaleen keskipisteestä reunakäyrän näytteistettyihin pisteisiin ja muodostamalla näin etäisyysvektori (Wäldchen ja Mäder 2017). Näin saatu vektori on kuitenkin herkkä häiriöille eikä kovin tehokas vertailtaessa kahden eri kappaleen käyrän täsmäävyyttä. Etäisyysvektoria voidaan kuitenkin muokata rotaatioinvariantimmaksi muodostamalla vektorin arvoista histogrammi tai laskemalla käyrälle Fourier-muunnos ja tarkastelemalla käyrää taajuustasossa (Wäldchen ja Mäder 2017). Eräs keino tarkastella reunakäyrää on laskea käyrän fraktaalidimensio, joka kertoo käyrän mutkikkuudesta tietyllä matkalla ja kuvastaa siten kappaleen sopivuutta omaan dimensioonsa (Wäldchen ja Mäder 2017). Menetelmää on hyödynnetty kukkia useammin lehden muotoon perustuvissa lajintunnistustutkimuksissa. Melko monimutkainen menetelmä on CSS (*curvature-scale space*), jossa tarkastellaan reunakäyrän pisteiden ympäristön kuperuutta ja koveruutta eri skaaloissa (Wäldchen ja Mäder 2017).

Värejä voidaan kuvata monenlaisissa väriavaruuksissa. Ganesan ym. (2010) kuvaavat värimallia ja väriavaruutta seuraavasti:

> Värimalli on abstrakti matemaattinen malli, joka kuvaa, miten värit voidaan esittää numeroista muodostetuissa monikoissa, jotka tyypillisesti koostuvat kolmesta tai neljästä väriä kuvaavasta komponentista. Väriavaruus saadaan, kun monikkoon liitetään tarkka kuvaus, miten monikon komponentit tulee tulkita.

Useimmiten vastaantuleva värimalli on digitaalikameroissa ja näytöissäkin käytössä oleva RGB, jossa päävärejä, ja siten monikossa edustettuina, ovat punainen, vihreä ja sininen (Loesdau, Chabrier ja Gabillon 2014). Kuvia voidaan suhteellisen helposti muuntaa värimallista toiseen, ja tilanteesta riippuu, mitä mallia kannattaa kulloinkin käyttää (Ganesan, Rajini ja Immanuvel Rajkumar 2010). Kukkien luokittelututkimuksissa on käytetty muun muassa HSV- (*hue-saturation-value*) ja Lab-värimalleja (Seeland ym. 2017). Kuvien väri-informaatiota on tiivistetty esimerkiksi histogrammeihin ja jakaumien tunnuslukuihin, jotka ovat laskennallisesti helppoja ja nopeita menetelmiä käytettäväksi myös reaaliaikaista tunnistusta vaativissa järjestelmissä (Wäldchen ja Mäder 2017). Muutamissa tutkimuksissa on hyödynnetty myös C-SIFT-deskriptoria (Wäldchen ja Mäder 2017), joka vastaa SIFT-menetelmää, mutta kuvaa muodon lisäksi väri-informaatiota (Abdel-Hakim ja Farag 2006).

Tekstuuria voidaan niinikään analysoida monella eri menetelmällä. Kukan pinnan tekstuuria on analysoitu konvolvoimalla kuvaa Gaborin suotimilla, hyödyntämällä Fourier-analyysia tai muodostamalla kappaleen pinnasta löydettyjen reunojen gradienttien suunnista histogrammeja (engl. *edge orientation histogram descriptor*) (Wäldchen ja Mäder 2017). Fraktaalidimensiota voidaan hyödyntää myös tekstuurin tutkimisessa: SFTA-menetelmä (*segmentation-based fractal texture analysis*) pilkkoo kuvan osasegmentteihin, ja kunkin segmentin reunakäyrästä lasketaan fraktaalidimensio (Wäldchen ja Mäder 2017).

2.3.2 Lab-väriavaruus

Sekä Lab- että HSV-värimallit on suunniteltu RGB-avaruutta paremmin vastaamaan ihmisen kykyä havainnoida värejä ja niiden eroja (Ganesan, Rajini ja Immanuvel Rajkumar 2010; Loesdau, Chabrier ja Gabillon 2014). Lab-mallissa kahden värin välinen euklidinen etäisyys vastaa ihmisen havaitsemaa värimuutosta (Ganesan, Rajini ja Immanuvel Rajkumar 2010). Lab-väriavaruus pystyy kuvaamaan kaikki ihmissilmän näkemät värit, ja sen perusideana on, että kaikki värit voidaan kuvata punaisen ja sinisen, punaisen ja keltaisen, vihreän ja sinisen sekä vihreän ja keltaisen yhdistelminä (Ganesan, Rajini ja Immanuvel Rajkumar 2010). Malli koostuu kolmesta toisiinsa nähden ortogonaalisesta koordinaattiakselista: L-akseli ilmaisee kirkkautta ja vastaa jotakuinkin ihmisen näkemystä kirkkauseroista, a-akseli koodaa vastaväriparia vihreä–punainen ja b-akseli koodaa vastaväriparia sininen–keltainen (Ganesan, Rajini

ja Immanuvel Rajkumar 2010). Lab-värimallin tavoin myös HSV-mallissa kromaattinen eli väriin liittyvä informaatio on erotettu omiksi komponenteikseen, värisävyksi ja saturaatioksi (Loesdau, Chabrier ja Gabillon 2014). HSV-mallin värisävy ja saturaatio ovat itse asiassa Lab-mallin värikoordinaatit napakoordinaatistoon muunnettuina, mikä tekee niiden arvoista Lab-mallin arvoja hankalammin verrattavissa olevia (Loesdau, Chabrier ja Gabillon 2014).

2.3.3 Momentit

Kuvamomentit kuvailevat kohteen muotoa todennäköisyysjakaumien muotoa kuvailevien tilastollisten momenttien tapaan. Kaksiulotteiset asteen $p+q$ geometriset momentit määritellään jatkuvalle kuvafunktiolle $f(x,y)$ seuraavasti (Hu 1962; Liao 1993):

$$M_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q f(x,y)dxdy,$$

missä $p,q = 0,1,2,\ldots$. Vastaavasti diskreettiarvoiselle kuvalle $I(x,y)$ momentit lasketaan kaavalla (Donchenko ja Golik 2013)

$$M_{ij} = \sum_x \sum_y x^i y^j I(x,y),$$

missä $i,j = 0,1,2,\ldots$.

Jos $f(x,y)$ oletetaan paloittain jatkuvaksi ja rajoitetuksi funktioksi, joka voi saada nollasta eroavia arvoja vain äärellisessä määrässä tason (x,y) pisteitä, niin kaikki asteen $p+q$, $p,q = 0,1,2,\ldots$, momentit ovat olemassa ja tällöin jokainen kuva voidaan kuvata yksilöllisesti äärettömällä määrällä momentteja (Hu 1962; Liao 1993). Käytännössä kuvadataa on kuitenkin mahdollista kuvata vain muutamilla alemman asteen momenteilla (Donchenko ja Golik 2013).

Harmaasävykuvan nollannen asteen momentti $M_{00} = \sum_x \sum_y I(x,y)$ kuvaa pikselien kirkkausarvojen jakaumaa kuva-alueella. Jos kirkkausarvot tulkitaan tiheyden kaltaiseksi suureeksi, nollannen asteen momentin voidaan ajatella kuvaavan kuva-alan massaa (Liao 1993). Vastaavasti binääriarvoisesta kuvasta laskettuna nollannen asteen momentti kuvaa kohteen pinta-alaa (Liao 1993). Jakamalla x ja y-suuntaiset ensimmäisen asteen momentit M_{10} ja M_{01} kuvan kokonaismassalla saadaan massakeskipiste $(\bar{x},\bar{y})$ (Liao 1993),

$$\bar{x} = \frac{M_{10}}{M_{00}} \text{ ja } \bar{y} = \frac{M_{01}}{M_{00}}.$$

Keskipisteen $(\bar{x},\bar{y})$ avulla voidaan määritellä keskeismomentit (Hu 1962; Liao 1993)

$$\mu_{ij} = \sum_x \sum_y (x-\bar{x})^i (y-\bar{y})^j I(x,y).$$

Ne ovat invariantteja kohteen siirrolle (Liao 1993).

Normalisoimalla keskeismomentit saavutetaan skaalainvarianttius (Donchenko ja Golik 2013):

$$\eta_{ij} = \frac{\mu_{ij}}{\mu_{00}^{(i+j+2)/2}}.$$

Normalisoiduista keskeismomenteista saadaan laskettua Hun momentit, jotka ovat siirron ja skaalauksen lisäksi invariantteja myös rotaatiolle (Hu 1962). Hun momentit lasketaan seuraavasti:

$$\begin{aligned}
\phi_1 &= \eta_{20} + \eta_{02} \\
\phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
\phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\
\phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\
\phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
&\quad + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
\phi_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\
\phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
&\quad - (\eta_{30} - 3\eta_{21})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2].
\end{aligned}$$

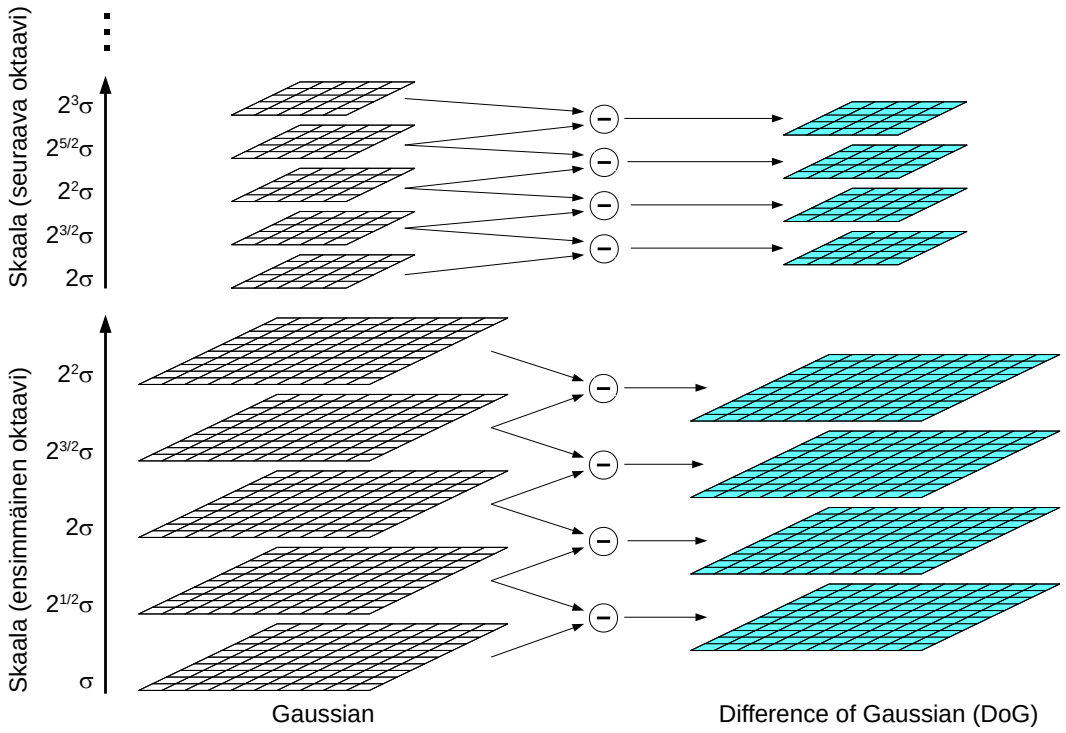
Seitsemäs Hun momentti pystyy erottelemaan toisistaan peilikuvat (Hu 1962). Ensimmäiselle Hun momentille taas on olemassa selkeä tulkinta hitausmomenttina (esim. Donchenko ja Golik 2013). Flusser (2000) osoitti kolmannen momentin olevan riippuvainen toisista momenteista. Flusser (2000) osoitti myös, ettei Hun invarianttien momenttien joukko ole täydellinen, vaan on olemassa lisää invariantteja momentteja, jotka eivät riipu Hun momenteista.

2.3.4 SIFT

Lowen (2004) kehittelemä SIFT-menetelmä löytää avainpisteitä ja muodostaa niille piirteet, jotka ovat invariantteja niin skaalan, rotaation kuin sijainninkin suhteen. SIFT-menetelmä koostuu kahdesta päävaiheesta. Ensimmäisessä vaiheessa kuvasta etsitään skaalainvariantteja ja suhteellisen vakaita avainpisteitä. Tyypillisesti avainpisteet ovat pieniä pistemäisiä tai nurkkamaisia kohteita, jotka eroavat kontrastiltaan muusta paikallisesta ympäristöstä. Skaalainvarianttius saavutetaan etsimällä avainpisteitä useissa eri skaaloissa. Vakaus tarkoittaa, että kuvasta löydetään samat avainpisteet, vaikka esimerkiksi kuvakulma tai valaistusolosuhteet hieman muuttuisivatkin. Toisessa vaiheessa avainpisteiden paikallinen ympäristö piirteistetään siten, että piirteet ovat invariantteja kohteen skaalalle, rotaatiolle ja sijainnille. Menetelmä tuottaa suuren määrän avainpisteitä ja piirteitä, jotka kattavat koko tarkasteltavan kuvan useissa eri skaaloissa.

SIFT-menetelmän ensimmäinen askel on skaala-avaruuden rakentaminen. Tämä tapahtuu alipäästösuodattamalla kuvaa inkrementaalisesti Gaussin suotimilla siten, että suodatuksen voimakkuus kasvaa jokaisella kerralla vakion k verran. Toisin sanoen kuvaa konvolvoidaan Gaussin funktioilla, joiden keskihajonnat eroavat toisistaan vakion k verran. Kahden perättäisen suodatetun kuvan erotus tuottaa DoG-kuvan (*Difference of Gaussians*), joka saa ääriarvoja reuna-alueilla ja tunnistaa siten kuvassa olevia reunoja. Skaalojen muodostaminen on kuvattu kuviossa 2. Käytännössä skaala-avaruus rakennetaan kuvaoktaavi kerrallaan. Kuvaoktaavit saadaan pienentämällä edellisen oktaavin kuvaa neljäsosaan, ja prosessia voidaan jatkaa niin pitkään kuin kuvakoko antaa myöten. Skaalat oktaavien sisällä rakennetaan siten, että ne kattavat tasavälisesti koko skaala-avaruuden. Vakio s määrää, kuinka monta skaalaa oktaavia kohden lasketaan ja täten samalla myös skaalatiheyden kaavalla $k = 2^{1/s}$. Jotta skaalat kattavat koko skaala-avaruuden, suodatettuja kuvia oktaavia kohden täytyy muodostaa $s+3$ kappaletta. Kuvion 2 esimerkissä $s = 2$, mutta Lowe (2004) suosittelee arvoa 3. Seuraavan oktaavin aloituskuva saadaan kätevästi näytteistämällä senhetkisen oktaavin kolmanneksi ylin kuva neljäsosakokoon. Pinon kolmanneksi ylintä kuvaa on suodatettu Gaussin suotimella, jonka keskihajonta on kaksinkertainen oktaavin ensimmäiseen kuvaan nähden.

Lowe (2004) suosittelee käytettäväksi ensimmäisen kuvan suodatuksessa keskihajonnan arvoa $\sigma = 1,6$. Usein kuvaa on kuitenkin jo esikäsittelyn aikana suodatettu, jolloin lisäsuoda-

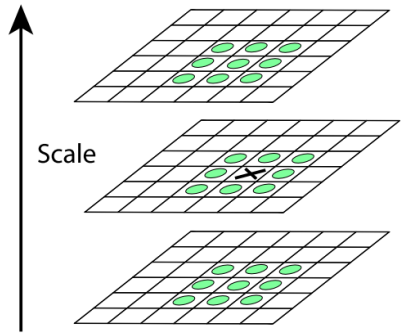


Kuvio 2: Skaala-avaruuden muodostaminen SIFT-algoritmissa. Kuvio on mukailtu Lowen (2004) artikkelissa olevasta kuviosta.

tuksen tarve on vähäinen. Tyypillisesti alkuperäisen kuvan kokoa myös kasvatetaan aluksi kaksinkertaiseksi, jotta löydetään enemmän avainpisteitä.

Oktaavien sisällä lasketaan perättäisten suodatettujen kuvien erotukset, jolloin kutakin oktaavia kohti saadaan $s+2$ DoG-kuvaa eli s kappaletta kuvapinossa keskellä sijaitsevia kuvia. Kuvapinossa keskellä sijaitsevien DoG-kuvien jokaista pikseliä verrataan sekä saman kuvan naapuripikseleihin että kuvaa edeltävän ja seuraavan kuvan naapuripikseleihin (kuvio 3). Jos pikselin arvo on jokaista naapuripikselin arvoa suurempi tai pienempi, pikseli valitaan potentiaaliseksi avainpisteeksi. Ääriarvon sijaintia voidaan edelleen tarkentaa interpoloimalla. Alkuperäinen Lowen algoritmi ei kuitenkaan sisältänyt interpolointiaskelta (Lowe 1999).

Potentiaalisten avainpisteiden määrää vähennetään poistamalla pisteet, joiden kontrasti eli absoluuttinen erotus DoG-kuvissa on pienempi kuin $0,03$. Kuvapikseleiden oletetaan saavan arvoja välillä $[0,1]$. Kontrastiraja-arvoa voi tarvittaessa pienentää tai suurentaa kuvan



Kuvio 3: SIFT-algoritmissa potentiaalisia avainpisteitä merkitsevät ääriarvot tunnistetaan vertailemalla DoG-kuvan jokaista pikseliä (ruksi) naapuripikseleihinsä (vihreät soikiot) perättäisten kuvien muodostamassa skaala-avaruudessa. Kuvio on Lowen (2004) artikkelista.

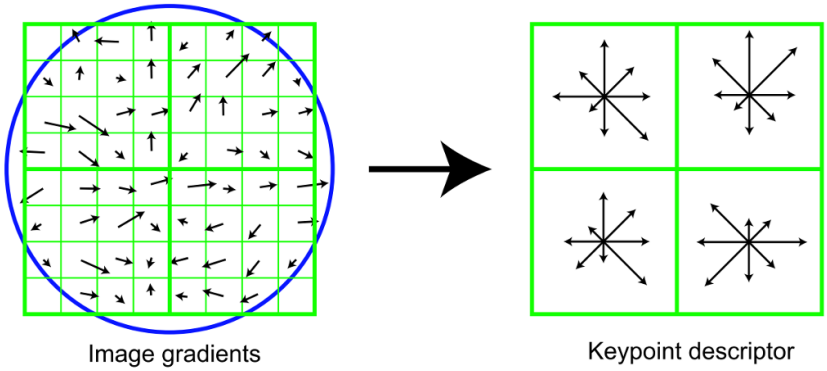
ominaisuuksista riippuen. Koska DoG-operaatio löytää ennen kaikkea reunoja, täytyy pisteiden joukosta poistaa vielä ne, joissa on havaittavissa vain yhdensuuntainen reuna. SIFT-algoritmissa tämä on toteutettu laskemalla Hessen matriisin jälki ja determinantti ja vertaamalla näistä saatua suhdelukua raja-arvoon. Hessen matriisi $\mathbf{H}$ sisältää DoG-kuvan osittaisderivaatat D_{xx}, D_{yy} ja D_{xy}, jotka voidaan laskea yksinkertaisesti naapuripikselien erotuksena. Hessen matriisin ominaisarvojen α ja β suhdeluku r kuvaa, onko kyseessä nurkka vai reuna. Mitä lähempänä ominaisarvot ovat toisiaan, sitä varmemmin kyseessä on nurkkapiste. Matriisin jäljellä $Tr(\mathbf{H})$ ja determinantilla $Det(\mathbf{H})$ taas on yhteys ominaisarvoihin, $Tr(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta$ ja $Det(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta$, joten ominaisarvojen suhdelukua voidaan tarkastella niiden perusteella. Lopullisessa raja-arvokaavassa matriisin jäljen ja determinantin suhdelukua verrataan suhdeluvusta r saatuun lukuun. Jos epäyhtälö

$$\frac{Tr(\mathbf{H})^2}{Det(\mathbf{H})} < \frac{(r+1)^2}{r}$$

on tosi, potentiaalinen avainpiste kelpuutetaan lopulliseksi avainpisteeksi. SIFT-algoritmissa suhdeluku r on yleensä 10.

Avainpisteen suunta määritetään suodatetusta kuvasta L, joka vastaa avainpisteen skaalaa. Avainpisteen ympäristön pikseleille (x,y) lasketaan gradientin suunta $\theta(x,y)$ ja voimakkuus $m(x,y)$ naapuripikseleiden erotuksiin perustuen:

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$
$$\theta(x,y) = tan^{-1}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y))).$$



Kuvio 4: Deskriptorien muodostaminen SIFT-algoritmissa. Avainpisteen ympäristön pikseleille määritetään gradienttien suunnat, joita painotetaan gradientin voimakkuuden ja etäisyysfunktion (sininen ympyrä) perusteella. Suunnista muodostetaan 8-luokkaiset histogrammit, joista edelleen muodostetaan deskriptorivektori. Kuvio on Lowen (2004) artikkelista.

Suunnista muodostetaan 36-luokkainen histogrammi. Siten jokainen histogrammin luokka käsittää kymmenen asteen lohkon. Histogrammia muodostettaessa niiden pikseleiden gradienttien suuntia, jotka ovat voimakkaita ja sijaitsevat lähimpänä avainpistettä, painotetaan eniten. Suuntahistogrammista tunnistetaan korkein huippu ja sen lisäksi etsitään vielä luokat, joiden korkeus on vähintään 80 % korkeimmasta huipusta. Nämä histogrammin luokat muodostavat avainpisteen suunnat. Jokainen suunta vastaa omaa avainpistettään. Avainpisteen suuntaa interpoloidaan tarkemmaksi sovittamalla paraabeli histogrammin huipun ympäristöön.

Algoritmin viimeisessä vaiheessa piirteistetään avainpisteen ympäristö. Avainpisteen ympäristön pikseleille lasketaan jälleen gradientin suunta ja voimakkuus pisteen skaalaa vastaavasta suodatetusta kuvasta (kuvio 4). Gradientiltaan voimakkaimmat suunnat ja avainpistettä lähimpänä olevat pikselit saavat jälleen suurimman painon — kuvion 4 vasemmanpuoleisessa kuvassa oleva ympyrä kuvaa etäisyyden painofunktiota. Pikseliympäristö jaetaan osa-alueisiin ja kunkin osa-alueen sisällä muodostetaan pikselien gradienttien suunnista 8-luokkaiset suuntahistogrammit (kuvio 4). Suuntainvarianttiuden saavuttamiseksi suunnat rotatoidaan avainpisteen pääsuunnan mukaan. Kuviossa 4 osa-alueita on neljä, mutta todellisuudessa avainpisteen ympärille muodostetaan 16 osa-aluetta, joista kukin sisältää 16 pikseliä. Osa-alueiden suuntahistogrammien arvot sijoitetaan vektoriin, jolloin saadaan

$8 * 16 = 128$ -alkioinen deskriptorivektori. Valaistusmuutosten vaikutusten minimoimiseksi vektorin pituus normalisoidaan ykköseksi ja kynnysarvon 0,2 ylittävät arvot tasataan kyseiseen arvoon, minkä jälkeen vektorin pituus normalisoidaan uudelleen. Arvo 0,2 määritettiin kokeellisesti. Käytäntö takaa sen, että histogrammin suurimmat voimakkuusarvot eivät enää dominoi niin voimakkaasti, kun deskriptorivektoreita vertaillaan toisiinsa.

2.3.5 *Bag-of-features* -menetelmä

Bag-of-features -menetelmä kehitettiin alunperin tekstidatan luokitteluun, jossa sitä kutsutaan nimellä *bag-of-words* (Zhang ym. 2013). Csurka ym. (2004) sovelsivat menetelmän käytettäväksi konenäön perusteella tapahtuvassa kuvien luokittelussa. He kutsuvat menetelmää artikkelissaan nimellä *bag-of-keypoints*. Menetelmän ensimmäisessä vaiheessa kuvasta tai sen osa-alueesta lasketaan paikallisten piirteiden deskriptorit. Deskriptorien tulee täyttää luvussa 2.1.2 mainitut vaatimukset eli olla invariantteja muun muassa rotaatiolle ja valoisuusvaihteluille, mutta pystyä kuitenkin samaan aikaan kuvaamaan riittävällä tasolla luokkien välisiä eroja. Seuraavaksi koulutuskuvien paikallisten piirteiden deksriptorit klusteroidaan; Csurka ym. (2004) käyttivät k-means-menetelmää. Klustereiden joukko muodostaa visuaalisen sanaston tai vektorin, jossa kukin klusterikeskus vastaa yhtä sanaa tai vektorin alkiota. Haasteena klusterointivaiheessa on määrittää sopiva sanaston pituus, joka havaitsee todelliset erot luokkien välillä mutta jättää satunnaiskohinan huomiotta — Csurka ym. (2004) määrittivät klustereiden määrän luokittelutarkkuuden perusteella. Tarkasteltavan kuvan jokainen paikallinen deskriptori määrätään lähimpään klusteriinsa, jolloin voidaan laskea lukumäärät, kuinka monta deskriptoria kuhunkin klusteriin sijoittuu. Näin saatu sanojen esiintymisen jakauma normalisoidaan, mistä muodostuu lopullinen *bag-of-features* -vektori. Csurka ym. (2004) sijoittivat tarkasteltavan kuvan kunkin deskriptorin vain yhteen lähimpään klusteriin, mutta kuten Zhang ym. (2013) mainitsevat, deskriptori voitaisiin sijoittaa myös useampaan kuin yhteen klusteriin.

3 Bayes-verkot

3.1 Yleistä Bayes-verkoista

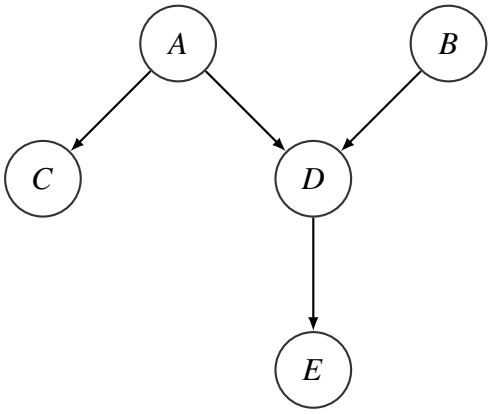
Bayes-verkot kuuluvat graafisiin malleihin, jotka yhdistävät verkkoteorian todennäköisyysteoriaan. Graafiset mallit tarjoavat kehyksen, jossa verkon rakenne kuvaa datan rakenteen muuttujien välisine yhteyksineen ja todennäköisyysteoria nivoo yhteen verkon osat yhtenäiseksi, toimivaksi malliksi (Jordan 1999). Niillä voidaan mallintaa epävarmuutta sisältäviä, monimutkaisia ilmiöitä, jotka voidaan verkon avulla jaotella pienempiin osiin (Jordan 1999). Bayes-verkkojen avulla voidaan esimerkiksi laskea ehdollisia todennäköisyyksiä tapahtumille ehdolla, että jotakin on havaittu (Charniak 1991).

Bayes-verkot ovat rakenteeltaan suunnattuja, syklittömiä verkkoja, joiden solmut ovat satunnaismuuttujia ja kaaret ilmentävät suoria riippuvuuksia muuttujien välillä (Charniak 1991). Kaaret voidaan tulkita myös kausaalisuhteiksi muuttujien välillä (Charniak 1991). Jos solmusta X_i lähtee suunnattu kaari solmuun X_j, on solmu X_j tällöin solmun X_i lapsi ja sen arvo riippuu solmun X_i arvosta. Solmu X_j on kuitenkin riippumaton muista verkon solmuista ehdolla (välittömät) vanhempansa. Kuviossa 5 on esitetty yksinkertainen Bayes-verkko, jonka rakenne on yhteneväinen Charniakin (1991) artikkelin esimerkkiverkon kanssa. Esimerkkiverkossa solmun E todennäköisyysjakauma voidaan täten muodostaa pelkästään solmun D todennäköisyysjakauman perusteella. Tämä vähentää huomattavasti verkon konstruoimiseen tarvittavien parametrien määrää (Charniak 1991). Formuloidaan Bayes-verkon sisältämät ehdolliset riippumattomuudet seuraavaksi matemaattisemmin.

Olkoot X_i, missä $i = 1, \ldots, n$, joukko satunnaismuuttujia. Niiden yhteistodennäköisyysjakauma $P(X_1, \ldots, X_n)$ määrittää muuttujien jakauman tyhjentävästi. Se voidaan faktoroida osiin seuraavasti (Charniak 1991):

$$P(X_1, \ldots, X_n) = P(X_1)P(X_2|X_1)\cdots P(X_n|X_1, \ldots, X_{n-1}).$$

Olkoot edelleen Q, R ja S joukko satunnaismuuttujia. Muuttujien Q ja R määritellään olevan toisistaan riippumattomia ehdolla S, jos $P(Q|R,S) = P(Q|S)$ aina kun $P(R,S) > 0$ (Friedman, Geiger ja Goldszmidt 1997). Määritellään Bayes-verkon solmujoukko Π_{X_i} solmun X_i van-



Kuvio 5: Esimerkki Bayes-verkosta.

hemmiksi. Tällöin edellä esitetty yhteistodennäköisyysjakauma voidaan ehdollisten riippumattomuuksien kaavaa käyttäen muuntaa muotoon (Friedman, Geiger ja Goldszmidt 1997)

$$P(X_1,\ldots,X_n) = \prod_{i=1}^{n} P(X_i|\Pi_{X_i}).$$

Kuvion 5 esimerkkiverkon tapauksessa yhteistodennäköisyysjakauma

$$P(A,B,C,D,E) = P(A)P(B|A)P(C|A,B)P(D|A,B,C)P(E|A,B,C,D)$$

muuntautuu verkon sisältämät ehdolliset riippumattomuudet huomioon ottaen muotoon

$$P(A,B,C,D,E) = P(A)P(B)P(C|A)P(D|A,B)P(E|D).$$

Täten riittää määrätä verkon juurisolmuille prioritodennäköisyydet ja ei-juurisolmuille ehdolliset todennäköisyydet ehdolla välittömät vanhempansa (Charniak 1991).

Bayes-verkossa minkä tahansa kiinnostuksen kohteena olevan muuttujan Y arvon todennäköisyys ehdolla havaittu muuttuja X saadaan laskettua Bayesin teoreeman peruskaavasta (Cowell 1999)

$$P(Y|X) = \frac{P(X,Y)}{P(X)} = \frac{P(X|Y)P(Y)}{P(X)}. \tag{3.1}$$

Muuttujan $P(Y)$ posterioritodennäköisyys $P(Y|X)$ saadaan siten päivittämällä muuttujan priorijakaumaa $P(Y)$ havaitun datajoukon uskottavuudella $P(X|Y)$, joista jälkimmäinen termi kuvastaa todennäköisyyttä saada tietynlainen otos muuttujan Y arvon ollessa kiinnitetty (Cowell 1999). Jakaja $P(X)$ normalisoi jakauman todennäköisyysjakaumaksi.

Jos kuvion 5 esimerkkiverkosta haluttaisiin laskea todennäköisyys muuttujan B arvolle b, kun on havaittu arvot $C = c$ ja $E = e$, saataisiin yhtälö

$$P(B = b|C = c, E = e) = \frac{\sum_{A,D} P(A, B = b, C = c, D, E = e)}{\sum_{A,B,D} P(A, B, C = c, D, E = e)}$$
$$= \frac{\sum_{A,D} P(A)P(B = b)P(C = c|A)P(D|A, B = b)P(E = e|D)}{\sum_{A,B,D} P(A)P(B)P(C = c|A)P(D|A, B)P(E = e|D)}.$$

Todennäköisyyksiä laskettaessa joudutaan siten marginalisoimaan eli summaamaan yli niiden muuttujien, joista ei olla kiinnostuneita tai joista ei ole havaintoja (Myllymäki ja Tirri 1998).

3.2 Päättely Bayes-verkoissa

3.2.1 Päättely erilaisissa tilanteissa

Pienien Bayes-verkkojen tapauksessa muuttujien todennäköisyydet ovat laskettavissa suoraan yhteistodennäköisyysjakaumasta summaamalla yli muuttujien, jotka eivät ole tarkastelun kohteena, kuten luvussa 3.1 havaittiin. Muuttujien määrän kasvaessa näiden marginaalijakaumien laskeminen käy kuitenkin hankalaksi, jopa mahdottomaksi. Itse asiassa ehdollisten todennäköisyysjakaumien laskeminen on NP-täydellinen ongelma monipolkuisilla verkoilla, joilla jonkin kahden solmun välillä on enemmän kuin yksi suuntaamaton polku (Myllymäki ja Tirri 1998). Laskettaessa marginaalisummia suoraviivaisesti koko yhteistodennäköisyysjakaumasta joudutaan myös laskemaan samoja summia turhaan moneen kertaan (Ankan ja Panda 2015). Jos esimerkiksi kuvion 5 mukaisessa Bayes-verkossa kaikki muuttujat olisivat binäärisiä ja haluttaisiin laskea todennäköisyys muuttujan E arvolle 1, termi $P(A = 1)P(C = 1|A = 1)$ laskettaisiin neljä kertaa eli kaikilla muuttujien B ja D mahdollisilla arvokombinaatioilla. Marginaalijakaumien laskemisen ongelmaan on kehitetty erilaisia algoritmeja, joista tärkeimpiä ovat viestinvälitysalgoritmi (engl. *belief propagation*) ja muuttujien eliminointi (engl. *variable elimination*) (Ankan ja Panda 2015). Yleisperiaate molemmissa algoritmeissa on sama eli laskennan jakaminen pienempiin yksiköihin (Ankan ja Panda 2015).

Viestinvälitys- ja muuttujien eliminointi -algoritmit on esitetty tässä tutkimuksessa päättelytilanteessa, jossa halutaan laskea tarkastelun kohteena olevan muuttujan tietyn havain-

toarvon todennäköisyys tai yleisemmässä tapauksessa muodostaa muuttujan posterioritodennäköisyys. Toinen yleinen päättelytehtävä on löytää havaintoarvo, joka maksimoi tarkasteltavan muuttujan posterioritodennäköisyyden (Dechter 1999). Ainoa muutos, joka sekä viestinvälitys- että muuttujien eliminointi -algoritmeihin tarvitaan, on summafunktion sijaan käyttää maksimointifunktiota (Dechter 1999; Ankan ja Panda 2015).

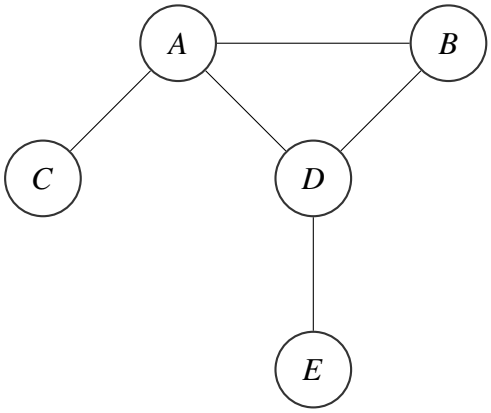
Esitettävät päättelyalgoritmit tarjoavat eksaktin ratkaisun. Verkon rakenteen monimutkaistuessa eksaktit ratkaisut käyvät kuitenkin laskennallisesti mahdottomiksi (Ankan ja Panda 2015). Bayes-verkoille onkin olemassa myös laskentaa yksinkertaistavia, approksimoivia päättelyalgoritmeja. Ne sopivat myös tilanteisiin, joissa ei ole tarpeen tietää muuttujien arvojen tarkkoja todennäköisyyksiä (Ankan ja Panda 2015). Approksimoivat päättelyalgoritmit sivuutetaan kuitenkin tässä tutkimuksessa, koska niitä ei ole tarkoitus käyttää tutkimuksen tuloksia laskettaessa.

3.2.2 Viestinvälitysalgoritmi

Puumallisessa verkossa kunkin solmuparin välillä on vain yksi mahdollinen reitti. Viestinvälitysalgoritmi perustuu siihen, että puumallisessa verkossa solmut jakavat verkon erillisiin osiin, jotka voidaan kukin käsitellä erikseen (Myllymäki ja Tirri 1998). Viestinvälitysalgoritmissa tällaista puuta kutsutaan liittymäpuuksi (engl. *junction tree* tai *clique tree*) (Myllymäki ja Tirri 1998; Ankan ja Panda 2015). Mistä tahansa Bayes-verkosta voidaan rakentaa liittymäpuu seuraavalla algoritmilla (Cowell 1999; Myllymäki ja Tirri 1998).

1. Yhdistetään kaarella vanhemmat, joilla on yhteinen lapsi.
2. Muutetaan kaikki kaaret suuntaamattomiksi, jolloin vaiheiden 1 ja 2 jälkeen saadaan niin sanottu moraalinen verkko. Kuviossa 6a on esitetty kuvion 5 esimerkkiverkon moraalinen verkko.
3. Kolmioidaan moraalinen verkko eli lisätään kaaria solmujen välille siten, että kaikki syklit koostuvat enintään kolmesta solmusta. Esimerkkiverkossa kaikki syklit koostuvat jo alkujaan enintään kolmesta solmusta, joten lisää kaaria ei tarvita.

4. Tunnistetaan verkon klikit eli verkon maksimaaliset aliverkot, joissa kaikki solmut ovat liittyneenä toisiin solmuihin. Esimerkkiverkossa klikkejä ovat solmujen joukot (A,C), (A,B,D) ja (D,E).
5. Yhdistetään kunkin klikin solmut uudeksi muuttujaksi ja rakennetaan näin saatujen uusien muuttujien pohjalta liittymäpuu. Puun solmusta toiseen voi olla polku vain, jos molemmissa solmuissa on ainakin yksi sama alkuperäinen muuttuja. Kuviossa 6b on kuvattu kuvion 5 esimerkkiverkon liittymäpuu.



(a) Esimerkkiverkon moraalinen verkko.

(b) Esimerkkiverkon liittymäpuu.

Kuvio 6: Liittymäpuun muodostaminen kuvion 5 esimerkkiverkosta.

Liittymäpuun rakentaminen ei yleensä ole yksikäsitteistä ja toisaalta klikkien koko määrää laskennan kompleksisuuden (Cowell 1999). Ankan ja Panda (2015) esimerkiksi hyödyntävät kaarien etsimisessä suurimman virittävän puun algoritmia (engl. *maximum spanning tree*), joka löytää kaarien painojen maksimisumman kaaren painon kuvatessa sen yhdistämien solmujen yhteisten muuttujien määrää.

Liittymäpuun laskentayksikkönä on solmu. Koska naapurisolmut sisältävät yhteisiä muuttujia, välittyy näiden muuttujien mukana tietoa solmusta toiseen. Jos oltaisiin edelleen kiinnostuneita esimerkkiverkon muuttujasta E, täytyisi ensin välittää viesti solmusta (A,C) solmuun (A,B,D) ja sieltä edelleen uusi viesti solmuun (D,E) (kuvio 6b). Ankanin ja Pandan (2015)

kirjan merkintöjä seuraten viesti τ solmusta C_j solmuun C_i on yleisessä muodossa

$$\tau_{j\to i} = \sum_{C_j - S_{i,j}} \psi_j \prod_{k \in naapuri(j) - \{i\}} \tau_{k\to j},$$

missä ψ_j on solmun C_j yhteistodennäköisyysjakauma ja $\tau_{k\to j}$ on viesti solmusta C_k solmuun C_j. Kaavan summalauseke kertoo, että solmun C_j yhteistodennäköisyysjakaumasta eli potentiaalista ψ_j lasketaan marginaalijakauma niiden muuttujien joukolle $S_{i,j}$, jotka ovat yhteisiä solmuille C_i ja C_j. Toisin sanoen lausekkeessa summataan yli solmun C_j kaikkien niiden muuttujien, jotka eivät kuulu joukkoon $S_{i,j}$. Tulolauseke taas kertoo, että ennen kuin viesti $\tau_{j\to i}$ voidaan muodostaa, täytyy solmun C_j saada viestit kaikilta sen naapurisolmuilta paitsi niiltä, joille se on viestiä lähettämässä. Liittymäpuun viestiketjussa olevat solmut kokoavat näin ollen aina edeltävien solmujen informaation ja täydentävät sitä omalla informaatiollaan, kunnes päästään päätesolmuun, jonka muuttujan todennäköisyys on kiinnostuksen kohteena. Viestiketjun viimeiselle solmulle i voidaan määritellä uskottavuus β_i kaavalla

$$\beta_i = \psi_i \prod_{k \in naapuri(i)} \tau_{k\to i},$$

josta saadaan laskettua kiinnostuksen kohteena olevan muuttujan todennäköisyysjakauma summaamalla niiden muuttujien yli, joista ei olla kiinnostuneita. Liittymäpuun viestiketjussa lähdetään liikkeelle lehtisolmuista, joiden viesti muodostetaan marginalisoimalla solmun potentiaali ψ_j, joka on samalla solmun uskottavuus β_j.

Annetaan viestinvälitysalgoritmista esimerkki liittyen kuvion 5 esimerkkiverkkoon. Oletetaan, että tarkastellaan edelleen muuttujaa E. Verkon liittymäpuu (kuvio 6b) on sen verran yksinkertainen, että ainoa vaihtoehto on lähteä liikkeelle lehtisolmusta (A,C). Koska muuttuja A on yhteinen solmuille (A,C) ja (A,B,D), se toimii viestinviejänä näiden solmujen välillä. Ensimmäinen viesti saadaan siten laskemalla solmun (A,C) yhteistodennäköisyysjakaumasta muuttujan A marginaalijakauma $\sum_C P(A,C)$. Seuraavana ketjussa on solmu (A,B,D). Koska tällä solmulla ei ole muita naapurisolmuja kuin (A,C) lukuunottamatta solmua (D,E), jolle se on viestiä välittämässä, sen tarvitsee odottaa viestiä vain solmulta (A,C) ennen viestin edelleenlähetystä. Koska muuttuja D on yhteinen solmuille (A,B,D) ja (D,E), se toimii viestinviejänä näiden solmujen välillä. Nyt viesti muodostuu solmuun (A,B,D) saapuneesta viestistä $\sum_C P(A,C)$ ja solmun (A,B,D) yhteistodennäköisyysjakaumasta lasketusta muuttu-

jan D marginaalijakaumasta $\sum_{A,B} P(A,B,D)$. Viesti on täten $\sum_{A,B} P(A,B,D) \sum_C P(A,C)$. Lopulta solmussa (D,E) voidaan laskea muuttujan E todennäköisyysjakauma, joka on solmuun saapuneiden viestien ja solmun (D,E) yhteistodennäköisyysjakaumasta lasketun muuttujan E marginaalijakauman tulo $\sum_D P(D,E) \sum_{A,B} P(A,B,D) \sum_C P(A,C)$. Luettaessa lauseketta oikealta vasemmalle voidaan nähdä, että lausekkeessa summataan yksi tai kaksi kerrallaan ulos ne muuttujat, joista ei olla kiinnostuneita, ja lopulta jäljelle jää vain muuttuja E. Solmujen yhteistodennäköisyysjakaumat saadaan alkuperäisestä Bayes-verkosta. Esimerkiksi tässä tapauksessa solmun (A,B,D) yhteistodennäköisyysjakauma on $P(D|A,B)P(A)P(B)$.

Liittymäpuu voidaan myös kalibroida kuljettamalla viestit molempiin suuntiin (Ankan ja Panda 2015). Kalibroidulle liittymäpuulle pätee, että naapurisolmujen yhteisen muuttujan marginaalijakaumat ovat samat molemmissa solmuissa. Kalibroidun puun avulla voidaan laskea minkä tahansa muuttujan todennäköisyys tarvitsematta käydä läpi uudelleen viestinvälitysalgoritmia.

3.2.3 Muuttujien eliminointi

Muuttujien eliminointi -algoritmin ovat kuvanneet muun muassa Dechter (1999) sekä Ankan ja Panda (2015). Algoritmi perustuu summaustermien uudelleenjärjestelyyn ryppäiksi, jonka myötä verkosta voidaan vähentää aina yksi muuttuja kerrallaan. Algoritmissa valitaan eliminoitava muuttuja, tunnistetaan muuttujat, joihin eliminoitava muuttuja on liittyneenä, ja muodostetaan näiden pohjalta ryppäälle eli aliverkolle yhteistodennäköisyysjakauma. Eliminoitava muuttuja poistetaan ryppäästä summaamalla yhteistodennäköisyysjakaumaa yli sen arvojoukon. Näin saadaan muodostettua poistetun muuttujan tilalle uusi faktori, joka koostaa jäljellejääneiden muuttujien marginaalijakaumat kyseisessä aliverkossa. Jatkossa yhteistodennäköisyyttä laskettaessa käytetään tätä uutta faktoria. Eliminoitu muuttuja myös poistetaan verkosta. Jos uuteen faktoriin sisältyy useampi kuin yksi muuttuja, täytyy verkkoon piirtää ylimääräiset kaaret muuttujien välille kuvaamaan niiden välistä yhteyttä.

Algoritmi selkeytynee esimerkillä. Samantyyppinen esimerkki mutta eri verkkoon perustuen on esitetty Dechterin (1999) artikkelissa. Oletetaan, että haluttaisiin laskea esimerkkiverkon

(kuvio 5) muuttujan E arvojen todennäköisyyksiä

$$P(E) = \sum_A \sum_C \sum_B \sum_D P(A)P(C|A)P(B)P(D|A,B)P(E|D).$$

Valitaan ensimmäiseksi eliminoitavaksi muuttujaksi B. Näin saadaan muodostettua uusi faktori $\tau_B(D,A) = \sum_B P(D|A,B)P(B)$ ja poistettua verkosta solmu B (kuvio 7a):

$$\begin{aligned} P(E) &= \sum_A \sum_C \sum_D P(A)P(C|A)P(E|D) \sum_B P(B)P(D|A,B) \\ &= \sum_A \sum_C \sum_D P(A)P(C|A)P(E|D)\tau_B(D,A). \end{aligned}$$

Eliminoidaan seuraavaksi muuttuja C, jolla itse asiassa ei ole vaikutusta muuttujaan E, jolloin summatermistä tulee 1 (kuvio 7b):

$$P(E) = \sum_A \sum_D P(A)P(E|D)\tau_B(D,A) \sum_C P(C|A) = \sum_A \sum_D P(A)P(E|D)\tau_B(D,A).$$

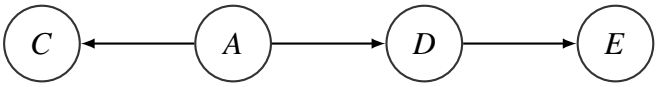
Eliminoidaan seuraavaksi muuttuja A (kuvio 7c):

$$P(E) = \sum_D P(E|D) \sum_A P(A)\tau_B(D,A) = \sum_D P(E|D)\tau_A(D).$$

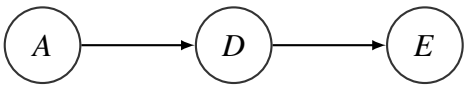
Muuttujan D eliminoinnin jälkeen saataisiin selville kiinnostuksen kohteena ollut muuttujan E marginaalijakauma. Jos esimerkissä olisi eliminoitu muuttuja A ennen muuttujaa C, olisi verkkoon pitänyt piirtää solmujen C ja D välille lisäkaari (kuvio 8). Tällöin olisi saatu

$$P(E) = \sum_C \sum_D P(E|D) \sum_A P(A)P(C|A)\tau_B(D,A) = \sum_C \sum_D P(E|D)\tau_A(C,D).$$

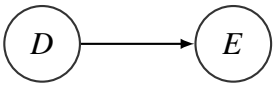
Kuten esimerkistäkin käy ilmi, muuttujien eliminointijärjestys voidaan valita usealla eri tavalla. Algoritmin laskennalliseen vaativuuteen vaikuttaa muodostettavan faktorin muuttujien määrä eli aliverkon koko, josta marginaalisummia lasketaan (Dechter 1999; Ankan ja Panda 2015). Faktorien muuttujien määrään taas vaikuttaa se, missä järjestyksessä muuttujia eliminoidaan ja minkä solmujen välille joudutaan tämän seurauksena piirtämään ylimääräisiä kaaria. Parhaan eliminointijärjestyksen löytäminen on kuitenkin NP-täydellinen ongelma (Ankan ja Panda 2015). Eräs melko hyvin toimiva tekniikka eliminointijärjestyksen määrittämiseen on ahne menetelmä, jossa jokaisella askeleella määritetään kustannusfunktioon



(a) Esimerkkiverkko muuttujan B poistamisen jälkeen.

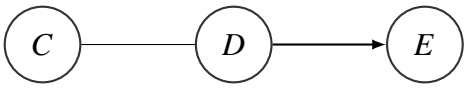


(b) Esimerkkiverkko muuttujan C poistamisen jälkeen.



(c) Esimerkkiverkko muuttujan A poistamisen jälkeen.

Kuvio 7: Muuttujien eliminoiminen kuvion 5 esimerkkiverkosta, kun eliminointijärjestyksenä on B, C, A ja D.



Kuvio 8: Kuvion 5 esimerkkiverkko, jos verkosta olisi eliminoitu muuttujat järjestyksessä B ja A.

perustuen senhetkinen paras ratkaisu (Ankan ja Panda 2015). Kustannusfunktio voi olla esimerkiksi muuttujaehdokkaan naapurien määrä verkossa tai niiden lisäkaarien määrä, jotka joudutaan piirtämään, jos kyseinen muuttuja eliminoidaan (Ankan ja Panda 2015).

Havaitut muuttujat kannattaa käsitellä algoritmissa poikkeavalla tavalla. Havaitun muuttujan tapauksessa uuden faktorin muodostamisen sijaan on tehokkaampaa sijoittaa havaittu arvo erikseen kuhunkin parhaillaan käsiteltävän ryppään todennäköisyysfunktioon ja sijoittaa nämä todennäköisyysfunktiot sellaisinaan myöhemmin käsiteltäviin ryppäisiin (Dechter 1999). Jos esimerkiksi edellisessä esimerkissä olisi havaittu muuttujan B arvo, ei olisi ollut tarpeen muodostaa kaksiulotteista faktoria $\tau_B(D,A)$. Laskentaa voidaan yksinkertaistaa myös tunnistamalla ryppäät, joissa todennäköisyys summautuu ykköseksi (Dechter 1999). Jos verkolle on määrätty topologinen järjestys, jossa vanhempisolmut edeltävät lapsisolmujaan ja kyselyn kohteena oleva solmu on ylimpänä, voidaan sivuuttaa rypäs, joka ei sisällä havaittua tai kyselyn kohteena olevaa muuttujaa tai uutta muodostettua faktoria (Dechter 1999).

3.3 Bayes-verkkojen parametrien ja rakenteen estimointi

Sopivaa mallia valittaessa käytetään yleensä jonkinlaista sopivuuskriteeriä, joka mittaa mallin sopivuutta aineistoon ja mahdolliseen prioritietoon (Heckerman 1999). Bayesiläinen lähestymistapa parametrien ja verkkorakenteen estimointiin on yhdistää prioritieto parametreista tai rakenteesta datan tarjoamaan informaatioon ja pyrkiä näin maksimoimaan parametrien tai rakenteen posterioritodennäköisyysfunktioita (Myllymäki ja Tirri 1998; Heckerman 1999).

3.3.1 Parametrien estimointi

Oletetaan, että mallin rakenne M, eli se osa mallista, joka kuvaa, miten muuttujasolmat ovat sijoittuneet toisiinsa nähden, on annettu valmiiksi. Seuraavana askeleena mallin rakennuksessa on estimoida malliin parhaiten sopivat parametrit θ. Mallin parametrit määräävät kunkin muuttujan todennäköisyysjakauman. Esimerkiksi normaalijakauman tapauksessa muuttujan todennäköisyysjakauman parametrijoukko koostuu keskiarvosta ja varianssista. Diskreeteillä muuttujilla kukin yksittäinen parametri θ_{ijk} taas kuvaa muuttujan X_i arvon k todennäköisyyden, kun muuttujasolmun vanhemmat ovat arvokombinaatiossa j. Jos esimerkiksi muuttujalla olisi kaksi binääriarvoista vanhempaa, ne voisivat olla $2^2 = 4$ erilaisessa arvokombinaatiossa.

Malliparametrin ehdollinen posterioritodennäköisyys $P(\theta|D,M)$ ehdolla havaittu aineisto D ja mallin rakenne M saadaan suoraviivaisesti soveltamalla Bayesin kaavaa 3.1 (Myllymäki ja Tirri 1998):

$$P(\theta|M,D) = \frac{P(D|\theta,M)P(\theta|M)}{P(D|M)} \propto P(D|\theta,M)P(\theta|M). \tag{3.2}$$

Kaavassa $P(\theta|M)$ on parametrin prioritodennäköisyys mallirakenteen ollessa tunnettu ja $P(D|\theta,M)$ on havaintoaineiston uskottavuus (Myllymäki ja Tirri 1998). Jakaja $P(D|M)$ on skaalaustermi.

Tiettyjen jakaumaoletusten vallitessa lauseke 3.2 yksinkertaistuu. Itse asiassa parametrin posterioritodennäköisyyden maksimoivan estimaatin $\hat{\theta}_{ijk}$ lauseke voidaan kirjoittaa suljetussa muodossa (Heckerman 1999; Myllymäki ja Tirri 1998). Oletetaan, että puuttuvia ha-

vaintoja ei ole, parametrit ovat riippumattomia toisistaan ja niiden priorijakaumat noudattavat Dirichlet-jakaumaa $\theta_{ijk} \sim Dir(\alpha_{ijk})$ (Heckerman 1999). Oletetaan edelleen, että data on otos jakaumasta, joka noudattaa multinomiaalijakaumaa; toisin sanoen satunnaismuuttuja X_i on diskreetti ja voi saada arvon joukosta, jossa on yhteensä r_i erilaista arvoa (Heckerman 1999). Tällöin voidaan osoittaa, että tarkimman estimaatin parametrille θ_{ijk} antaa kaava

$$\hat{\theta}_{ijk} = \frac{\alpha_{ijk} + N_{ijk}}{\sum_{k=1}^{r^i} \alpha_{ijk} + \sum_{k=1}^{r^i} N_{ijk}}, \tag{3.3}$$

missä α_{ijk} on parametrin θ_{ijk} priorijakauman parametri ja N_{ijk} on niiden tapausten lukumäärä, jossa muuttujan X_i arvo on k ja muuttujasolmun vanhemmat ovat arvokombinaatiossa j (Heckerman 1999; Myllymäki ja Tirri 1998). Jos prioritietoa ei ole käytettävissä tai tiedetään priorijakauman olevan tasainen, voidaan asettaa $\alpha_{ijk} = 1$ (Myllymäki ja Tirri 1998).

On hyödyllistä olettaa parametrin θ noudattavan juuri Dirichlet-jakaumaa laskennallisista syistä, sillä Dirichlet-jakauma on multinomiaalijakauman konjugaattipriori (Heckerman 1999). Tämä tarkoittaa, että parametrin θ priorijakauman noudattaessa Dirichlet-jakaumaa ja aineiston ollessa otos multinomiaalijakaumasta myös parametrin θ posteriorijakauma noudattaa Dirichlet-jakaumaa (Heckerman 1999). Tästä seuraa edelleen se, että priorijakaumaa voidaan päivittää helposti havaintotiedolla ja laskea parametrille uusi estimaatti yksinkertaisella kaavalla 3.3.

Kaava 3.3 voidaan johtaa laskemalla parametrin θ_{ijk} odotusarvo posteriorijakaumasta datan ja mallistruktuurin ollessa kiinnitetty (Heckerman 1999). Suurilla otoksilla näin saatu parametri lähestyy suurimman uskottavuuden parametria, joka saataisiin kaavasta 3.3 jättämällä pois priorijakaumaan viittaavat termit (Heckerman 1999). Datan sisältäessä puuttuvia havaintoja parametrien estimoinnissa voitaisiin hyödyntää esimerkiksi odotusarvon maksimointi -algoritmia (engl. *expectation-maximization algorithm*) ja käyttää termin N_{ijk} sijaan sen odotusarvoa (Heckerman 1999).

3.3.2 Rakenteen estimointi

Bayes-verkon rakenteen estimointi voidaan jakaa kahteen päämenetelmään (Tsamardinos, Brown ja Aliferis 2006). Sopivuuskriteereihin perustuvissa menetelmissä mallin rakenteen

yhteensopivuutta aineistoon arvioidaan erilaisten laskettavien mittojen perusteella. Rajoitteisiin perustuvissa menetelmissä verkon rakenne konstruoidaan havaittujen muuttujien välisiin ehdollisiin riippumattomuuksiin pohjautuen. On myös mahdollista yhdistää nämä kaksi menetelmää (Tsamardinos, Brown ja Aliferis 2006). Tutkimusongelma voi myös itsessään määrittää tai asettaa rajoitteita, millaisissa mahdollisissa rakenteissa muuttujasolmut voivat toisiinsa nähden sijaita (Kotsiantis 2007).

Sopivuuskriteereihin perustuvissa menetelmissä on mahdollista käydä läpi kaikki rakenteet ja laskea niille kriteerit, jos muuttujia on vain vähän. Yleensä sopivuuskriteeriin perustuvassa rakenteen estimoinnissa hyödynnetään kuitenkin heuristisia etsintäalgoritmeja, sillä verkkorakenteen estimointi on NP-täydellinen ongelma, jos muuttujasolmulla on enemmän kuin yksi vanhempi (Heckerman 1999). Eräs yksinkertainen etsintäalgoritmi on lokaalin maksimin löytävä ahne menetelmä, jossa valitaan ensin jokin alkuarvaus rakenteelle ja sen jälkeen joka iteraatiokierroksella muutetaan yhtä komponenttia kerrallaan. Kierroksen päätteeksi valitaan muutos, joka tuotti parhaimman sopivuuskriteerin (Heckerman 1999). Iteraatio pysäytetään, kun sopivuuskriteeri ei enää parane. Jotta päädyttäisiin todennäköisimmin globaaliin maksimiin lokaalin maksimin sijaan, voidaan valita useita satunnaisia alkuarvauksia ja suorittaa etsintäalgoritmi jokaiselle näistä (Heckerman 1999). Edelläkuvattua etsintäalgoritmia kutsutaan myös *hill climbing* -algoritmiksi ja sitä voidaan muunnella usealla eri tavalla liittyen muun muassa siihen, miten iteraatiokierroksen muutettavat komponentit valitaan — komponenttien valintaa voidaan esimerkiksi rajoittaa (Tsamardinos, Brown ja Aliferis 2006).

Parhaiten havaintoaineistoon D sopivaa mallirakennetta M estimoitaessa sopivuuskriteerinä käytetään mallirakenteen posterioritodennäköisyysfunktiota $P(M|D)$, joka pyritään maksimoimaan (Myllymäki ja Tirri 1998). Bayesin teoreemaa 3.1 soveltamalla rakenteen ehdollinen posterioritodennäköisyys saadaan laskettua kaavalla

$$P(M|D) = \frac{P(D|M)P(M)}{P(D)} \propto P(D|M)P(M), \tag{3.4}$$

missä $P(M)$ on mallirakenteen prioritodennäköisyys, $P(D|M)$ on marginaaliuskottavuus ja jakaja $P(D)$ on skaalaustermi (Myllymäki ja Tirri 1998). Marginaaliuskottavuus saadaan tavallisesti tilastotieteessä käytettävästä uskottavuudesta $P(D|\theta,M)$ painottamalla sitä para-

metrin θ prioritodennäköisyydellä ja integroimalla näin saatu lauseke $P(D|\theta,M)P(\theta|M)$ yli kaikkien struktuuria M vastaavien mallien (Myllymäki ja Tirri 1998; Heckerman 1999).

Oletetaan, että voimassa ovat samat oletukset, jotka pätivät johdettaessa kaava 3.3 parametriestimaattien laskemiseksi. Tällöin päästään eroon marginaaliuskottavuuden $P(D|M)$ laskemisessa tarvittavasta integraalista. Marginaaliuskottavuudelle voidaan tällöin johtaa kaava (Heckerman 1999; Myllymäki ja Tirri 1998)

$$P(D|M) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(\alpha_{ij}+N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk}+N_{ijk})}{\Gamma(\alpha_{ijk})}, \tag{3.5}$$

missä Γ on gammafunktio, n on muuttujien lukumäärä, q_i on muuttujasolmun X_i vanhempien mahdollisten arvokombinaatioiden lukumäärä, $\alpha_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}$ ja $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$. Muut termit ovat vastaavat kuin kaavassa 3.3.

Gammafunktion luonnollisille luvuille n pätee $\Gamma(n) = (n-1)!$ (Boas 1983, luku 11.3). Tällöin kaava 3.5 voidaan kirjoittaa muodossa

$$P(D|M) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{(\alpha_{ij}-1)!}{(\alpha_{ij}+N_{ij}-1)!} \prod_{k=1}^{r_i} \frac{(\alpha_{ijk}+N_{ijk}-1)!}{(\alpha_{ijk}-1)!}.$$

Tietyissä tilanteissa mallirakenteen posterioritodennäköisyyden laskemista voidaan edelleen helpottaa. K2-estimoinnissa malliparametrien θ priorijakaumien oletetaan olevan tasaisia, jolloin α_{ijk}=1[1] (Cooper ja Herskovits 1992). Myös erilaisille mallirakenteille oletetaan tasaiset priorit eli $P(M) = c$, missä c on jokin vakio (Cooper ja Herskovits 1992). Tällöin posterioritodennäköisyydestä saatava sopivuuskriteeri perustuu pelkästään marginaaliuskottavuuteen $P(D|M)$ (Myllymäki ja Tirri 1998). Täten K2-estimoinnissa pyritään maksimoimaan lauseke (Cooper ja Herskovits 1992)

$$P(M|D) \propto P(D|M) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{(\alpha_{ij}-1)!}{(\alpha_{ij}+N_{ij}-1)!} \prod_{k=1}^{r_i} N_{ijk}!.$$

Jos marginaaliuskottavuutta ei ole mahdollista laskea helposti suljetussa muodossa, sitä voidaan approksimoida eri tavoin (Heckerman 1999). Eräs approksimointimenetelmä on Bayesin informaatiokriteeri, BIC (Heckerman 1999):

$$log(P(D|M)) \approx log(P(D|\theta,M)) - \frac{d}{2} log(N),$$

1. Ks. pgmpy-dokumentaatio: estimators.K2Score

missä N on otosjoukko ja d on mallirakennetta M vastaavien parametrien lukumäärä. Myöskään BIC-menetelmässä ei hyödynnetä priori-informaatiota (Heckerman 1999). Kaavasta voidaan havaita, että sopivuuskriteeri tasapainoilee aineiston ja mallin yhteensopivuuden ja mallin monimutkaisuuden välillä (Myllymäki ja Tirri 1998).

Edellä kuvattuja sopivuuskriteerejä laskettaessa oletettiin kaikkien rakenteiden prioritodennäköisyyksien olevan yhtäsuuria. Sopiva rakenne löydetään tällöin pelkästään havaintoaineiston perusteella. On kuitenkin mahdollista hyödyntää asiantuntemusta erilaisten rakenteiden todennäköisyyksistä ja muodostaa näin rakenteille prioritodennäköisyydet (Heckerman 1999). Eräs vaihtoehto priorijakaumien konstruoimiseen on laskea posteriorijakauma simuloidulle datalle ja tasajakaumapriorille, ja soveltaa näin saatua todennäköisyyttä priorijakaumana varsinaisessa analyysissa (Heckerman 1999).

Rajoitteisiin perustuvassa verkkorakenteen estimointimenetelmässä muuttujien välisiä tilastollisia riippuvuuksia tarkastellaan erilaisten mittareiden avulla ja muuttujien keskinäinen sijainti verkossa määritellään näihin mittareihin perustuen (Myllymäki ja Tirri 1998; Ankan ja Panda 2015). Mittarit voivat olla tilastollisia tai informaatioteoreettisia riippumattomuusmittoja (Tsamardinos, Brown ja Aliferis 2006). Usein käytetty tilastollinen mittari muuttujien riippumattomuuden tutkimiseen on mitata muuttujien yhteisjakauman poikkeavuutta tilanteesta, jossa muuttujat oletetaan riippumattomiksi (Ankan ja Panda 2015). Eräs tällainen poikkeavuusmitta on χ^2-tunnusluku (Ankan ja Panda 2015)

$$d_{\chi^2}(D) = \sum_{x,y} \frac{(N[x,y] - N\hat{P}(x)\hat{P}(y))^2}{N\hat{P}(x)\hat{P}(y)}. \tag{3.6}$$

Kaavassa N on otoskoko, $N[x,y]$ on niiden datapisteiden lukumäärä, joissa satunnaismuuttujat X ja Y ovat saaneet arvot x ja y, ja $\hat{P}(x)$ ja $\hat{P}(y)$ ovat havaintoaineistosta estimoituja muuttujien arvojen todennäköisyyksiä. Jos muuttujat ovat riippumattomia, niille pätee $P(X,Y) = P(X)P(Y)$, jolloin summan osoittajat kaavassa 3.6 lähenevät nollaa.

Esimerkki informaatioteoreettisesta riippumattomuusmittarista on yhteinen informaatio, joka kuvastaa informaation määrää, jonka kaksi piirrevektoria jakavat — mitä suurempi yhteinen informaatio, sitä enemmän toisesta muuttujasta voidaan päätellä toisen muuttujan arvojen perusteella (Latham ja Roudi 2009). Tietyntyyppisten Bayes-verkkojen muodostami-

sessa käytetään yleisesti ehdollisen yhteisen informaation mittaria, joten esitetään se tässä yhteydessä. Kahden piirremuuttujan A_i ja A_j, $i \neq j$, ehdollinen yhteinen informaatio ehdolla luokkamuuttuja C lasketaan kaavalla (Friedman, Geiger ja Goldszmidt 1997)

$$I_P(A_i;A_j|C) = \sum_{a_i,a_j,c} P(a_i,a_j,c) log \frac{P(a_i,a_j|c)}{P(a_i|c)P(a_j|c)}. \tag{3.7}$$

Riippumattomuusmittaa verrataan raja-arvoon, joka määrää, onko lukema riittävän iso, jotta muuttujien voidaan olettaa olevan toisistaan riippuvia (Ankan ja Panda 2015). Menetelmä on siten hyvin herkkä raja-arvon valinnan suhteen.

3.4 Bayes-verkot tässä tutkimuksessa

Tutkimuksen Bayes-verkkomallit huomioivat eri tavoin piirremuuttujien välisiä tilastollisia riippuvuuksia. Naiivi Bayes-verkko olettaa piirremuuttujien olevan riippumattomia ehdolla luokkamuuttuja, ja piirteet ovat yhteydessä vain luokkamuuttujaan (esim. Friedman, Geiger ja Goldszmidt 1997). Naiivin Bayes-mallin rakenne on siten ennalta määrätty. Puu- tai metsärakenteella täydennetyissä naiivin Bayes-verkon muunnelmissa piirremuuttujien välille sallitaan riippuvuuksia yhdistämällä muuttujia toisiinsa suunnatuilla kaarilla (Friedman, Geiger ja Goldszmidt 1997; Lucas 2002). Malleissa jokaisella piirteellä on kuitenkin yhteys myös luokkamuuttujaan. Piirremuuttujien väliset riippuvuudet määräävät, minkä muuttujien välille lisäkaaria piirretään (Friedman, Geiger ja Goldszmidt 1997). Rakenteeltaan rajoittamattomat mallit kohtelevat luokkamuuttujaa kuten mitä tahansa piirremuuttujaa (Friedman, Geiger ja Goldszmidt 1997). Muuttujat voivat muodostaa hierarkkisia rakenteita, eikä kaikilla piirremuuttujilla ole välttämättä yhteyttä luokkamuuttujaan. Parhaiten aineistoon sopiva rakenne valitaan sopivuuskriteerin perusteella (Friedman, Geiger ja Goldszmidt 1997). Koostemuuttujamalleissa toisistaan voimakkaasti riippuvia piirteitä yhdistetään uudeksi muuttujaksi, joka on piirremuuttujien sijaan yhteydessä luokkamuuttujaan. Siten tässäkin mallissa piirremuuttujien väliset riippuvuustarkastelut määräävät mallin rakenteen. Tutkimuksen Bayes-verkkomallit kuvataan tarkemmin seuraavassa luvussa.

4 Menetelmien testaaminen ja tulokset

4.1 Tutkimusaineisto

Tutkimuksen aineiston muodosti osakokoelma Oxfordin yliopiston kukkakuva-aineistosta[1] (Nilsback ja Zisserman 2006). Yliopiston kuvakokoelmia on kaksi, jotka molemmat on luotu konenäkötutkimusta varten, ja ne ovat eräänlaisia benchmark-aineistoja tutkittaessa kukkien perusteella tapahtuvaa lajintunnistusta. Tutkielmassa käytettiin kokoelmista pienempää. Se sisältää 80 kuvaa kustakin 17 eri lajista, jotka kaikki ovat levinneisyydeltään yleisiä Iso-Britanniassa. Lajit ovat tunnettuja myös Suomessa joko luonnon- tai puutarhakasveina. Kuvat vaihtelevat niin skaalaltaan, valaistusolosuhteiltaan kuin kuvakulmiltaankin.

Koska kuvat ovat ominaisuuksiltaan hyvin vaihtelevia ja ne on kuvattu kasvien omassa elinympäristössä, kokoelmasta poimittiin viisi lajiluokkaa, joissa kohteet ovat kohtuullisesti erotettavissa taustasta niin kohteen värin kuin koonkin puolesta. Toisin sanoen tutkimukseen pyrittiin valitsemaan sellaisia kohteita tai lajeja, joiden voidaan olettaa olevan konenäöllisesti tunnistettavissa. Toisaalta lajit pyrittiin valitsemaan siten, etteivät ne olisi toisistaan erotettavissa pelkän väri-informaation perusteella. Edellämainitut ehdot täyttivät kuviossa 9 olevat päivänkakkara, voikukka, niittyleinikki, valkovuokko ja orvokki.

Kohteen löytämisen helpottamiseksi tutkimuskuvat rajattiin siten, että yksittäisessä kuvassa esiintyi vain yksi kohdeobjekti mahdollisimman tiiviisti ikkunoituna. Kuvat uudelleennäytteistettiin kokoon, jossa kuvan leveys oli 400 pikseliä. Esikäsittelynä kohinan ja pakkausartefaktien poistamiseksi kuvat suodatettiin Gaussin suotimella. Suodinikkunan kokona oli 9×9 pikseliä. Alipäästösuodatetut kuvat muunnettiin sekä harmaasävykuviksi, jotka normalisoitiin, että Lab-väriavaruuteen. Muunnokset toteutettiin OpenCV-kirjaston algoritmeilla.

4.2 Hyödynnetyt ohjelmistot

Ohjelmointikielenä toimi pääasiassa Python (v. 2.7.12/13). Perusdatankäsittelyssä, kuten datan muokkaamisessa sekä piirrevektoreiden, tunnuslukujen ja muiden suureiden laskennas-

1. http://www.robots.ox.ac.uk/~vgg/data/flowers/17/index.html

Kuvio 9: Luokiteltavat kasvilajit: päivänkakkara, voikukka, niittyleinikki, valkovuokko ja orvokki.

sa, hyödynnettiin paljon numpy-kirjastoa (v. 1.13.1). Konenäköön ja kuvadatan analysointiin käytettiin pääasiassa OpenCV-kirjaston algoritmeja (v. 2.4.13), mutta tarpeen tullen hyödynnettiin myös Matlabilla (v. R2014b) koodattuja toteutuksia. Data-analyysimenetelmissä ja kuvien piirtämisessä hyödynnettiin Pythonin koneoppimis- ja data-analyysikirjastoja sklearn (v. 0.19.0), scipy (v. 0.19.1) ja matplotlib (v. 2.0.2). Pandas-kirjastoa (v. 0.20.3) tarvittiin paikoin datan muokkaamisessa oikeaan muotoon. Graafimallien todennäköisyysjakaumien muodostamiseen, mallin rakentamiseen ja rakenteen estimointiin sekä lajiluokan ennustamiseen testivaiheessa käytettiin pgmpy-kirjastoa (v. 0.1.2)[2], joka on suunnattu erityisesti graafimalleille.

Tutkimusraporttiin on pyritty menetelmien kohdalle kirjaamaan niissä hyödynnetyt kirjastot ja algoritmitoteutukset. Jos samaa menetelmää on käytetty useassa vaiheessa, on menetelmätoteutukseen viitattu raportissa kuitenkin vain ensimmäisellä esiintymiskerralla. Jos esitetyn menetelmän tai algoritmin kohdalla ei ole kerrottu, millä se on toteutettu, on toteutus tehty itse Pythonin peruskirjastoja hyödyntäen.

4.3 Naiivin Bayes-luokittelijan keskus- ja reunaosamalli

4.3.1 Kuvien esikäsittely ja piirrevektoreiden muodostaminen

Luonnollinen tapa jakaa edestäpäin tarkasteltava kukka osiin on erottaa siitä kukkapohjuksen muodostama keskiosa ja terälehtien muodostama reunaosa. Jako ei tosin päde kaikkiin kasvilajeihin, sillä esimerkiksi tarkastelun kohteena olevista lajeista voikukan ja päivänkak-

2. http://pgmpy.org/

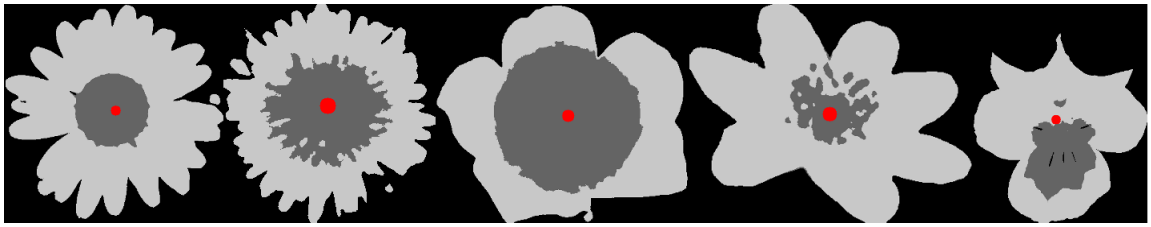
karan mykerökukinnot koostuvat sadoista pienistä kukista. Silmämääräisesti tarkasteltuna niistäkin on kuitenkin erotettavissa jonkinlainen keskiosa ja reunaosa.

Osajako muodostettiin jakamalla kohde segmentteihin pikselin värin ja kohteen keskipisteeseen lasketun suhteellisen etäisyyden perusteella. Kohteen keskipisteen määrittämistä varten harmaasävykuva kynnystettiin binääriarvoiseksi mustavalkokuvaksi, jossa kohde esitettiin valkoisilla pikseleillä ja tausta mustilla. Kun kynnystämiseen käytetään harmaasävykuvaa, kohde määräytyy pikselin valoisuusarvon perusteella. Raja-arvo määrää, luokitellaanko pikseli valoisaksi oletettuun kohteeseen vai tummaan taustaan. Tässä tapauksessa kuvien histogrammien oletettiin olevan kaksihuippuisia, ja OpenCV:n kynnystysalgoritmin[3] annettiin määrätä raja-arvo huippujen väliltä (Yousefi 2011). Osa kukista oli kuitenkin taustaa tummempisävyisiä. Tämän vuoksi kynnystystä korjattiin jälkikäteen ja tausta vaihdettiin kohteeksi, jos vähintään puolet kuvan reunapikseleistä oli valkoisia ja yhden sivun kaikki reunapikselit olivat valkoisia. Kynnystyksen jälkeen kohteeseen jääneitä pieniä reikiä ja avautumia pyrittiin vielä häivyttämään morfologisella sulkemisoperaatiolla[4] (Szeliski 2011, luku 3.3.2), ja lopuksi kaikki kohteen sisään jääneet reiät täytettiin[5]. Kohteen keskipiste laskettiin kohteeseen kuuluvien pikseleiden massakeskipisteenä.

Kuva segmentoitiin k-means-klusterointialgoritmilla[6] (Zaki ja Meira 2014, luku 13.1). Klusteroinnin piirremuuttujiksi valittiin Lab-värikanavien arvot ja pikselin euklidinen etäisyys kohteen keskipisteestä. Kaikki piirremuuttujat skaalattiin välille $[0, 1]$. Klustereiden määräksi valittiin kokeilujen perusteella neljä klusteria, koska se oli riittävä määrä sekä erottelemaan taustan kohteesta että jakamaan kohdetta osiin. Lähimpänä kohteen keskipistettä ollut klusteri muodosti kohteen keskiosan sellaisenaan. Kohteen reunaosan muodostivat toiseksi ja kolmanneksi lähimpänä keskipistettä olevat klusterit, joskin kolmanneksi lähimmästä klusterista otettiin mukaan vain pikselit, jotka kuuluivat kohteeseen myös kynnystyskuvassa. Esimerkki kukkien osajaosta objektikeskipisteineen on nähtävissä kuviossa 10.

Kukkia kuvattaessa erityisesti muoto ja väri ovat lajintunnistuksen kannalta oleellisia ja yleisesti tutkimuksissa tarkasteltuja ominaisuuksia (Wäldchen ja Mäder 2017). Siten myös tässä

3. Ks. OpenCV 2.4 -dokumentaatio: threshold
4. Ks. OpenCV 2.4 -dokumentaatio: morphologyEx
5. Ks. scipy-dokumentaatio: binary_fill_holes
6. Ks. OpenCV 2.4 -dokumentaatio: kmeans



Kuvio 10: Esimerkki kohteiden jaosta keski- ja reunaosaan sekä kohteen keskipiste (punainen piste). Yksilöt ovat samoja kuin kuviossa 9.

tutkimuksessa kukkaa päädyttiin kuvaamaan väri- ja muotopiirteillä. Ensimmäiseen malliin valittiin yksinkertaisia muoto- ja väripiirteitä: Hun momentit[7] (Hu 1962) sekä tilastollisia tunnuslukuja Lab-väriavaruuden a- ja b-kanavien arvoista. Lasketut tunnusluvut olivat keskiarvo, mediaani, keskihajonta sekä vinous- ja huipukkuusarvot.

Lasketuille piirteille suoritettiin jakaumien muotoja ja lajien välisiä eroja koskevia visuaalisia tarkasteluja, joiden perusteella keski- ja reunaosaa päädyttiin kuvaamaan yhdeksällä piirrevektorilla:

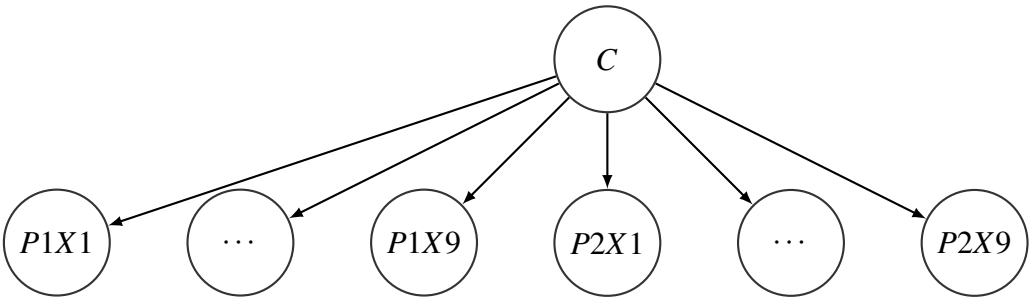
- Lab-väriavaruuden a- ja b-kanavien keskiarvot, keskihajonnat sekä jakaumien vinousarvot,
- Hun 2. ja 4. kuvamomentti logaritmiseen asteikkoon muunnettuina,
- keskus- ja reunaosan keskimääräiset euklidiset etäisyydet kohteen keskipisteestä suhteutettuna kuvan kokoon.

4.3.2 Mallin kuvaus ja rakentaminen

Naiivi Bayes-luokittelija on koneoppimisessa yleisesti käytetty yksinkertainen ja robusti luokittelumenetelmä (Friedman, Geiger ja Goldszmidt 1997). Siten oli luonnollista valita se ensimmäiseksi Bayes-verkkomalliksi myös tähän pro gradu -työhön. Analysoitavaa aineistoa vastaava naiivin Bayes-luokittelijan verkko on kuvattu kuviossa 11. Naiivin Bayes-verkon juuri- eli luokkasolmu kuvastaa luokkien todennäköisyyksiä. Piirrevektorisolmut, joiden perusteella luokittelu tapahtuu, ovat yhteydessä ainoastaan luokkasolmuun. Niiden oletetaan siten olevan toisistaan tilastollisesti riippumattomia ehdolla kohteen luokka (Friedman, Geiger

7. Ks. OpenCV 2.4 -dokumentaatio: HuMoments

ja Goldszmidt 1997). Kunkin luokan prioritodennäköisyys $P(c_i)$ muodostetaan yleensä luokan c_i suhteellisesta frekvenssistä (Zaki ja Meira 2014). Piirrevektorien todennäköisyydet ehdolla luokka eli uskottavuudet $P(x_1,\cdots,x_n|c_i)$ estimoidaan koulutusdatasta (Zaki ja Meira 2014). Tämän jälkeen testidatasta voidaan laskea luokan c_i posterioritodennäköisyys Bayeskaavalla $P(c_i|x_1,\cdots,x_n) \propto P(c_i)P(x_1,\cdots,x_n|c_i)$, joka piirrevektorien keskinäinen riippumattomuus huomioiden muokkautuu muotoon $P(c_i|x_1,\cdots,x_n) \propto P(c_i)\prod_{j=1}^{n} P(x_j|c_i)$. Testidata kuuluu luokkaan, jonka posterioritodennäköisyys saa suurimman arvon (Zaki ja Meira 2014).



Kuvio 11: Naiivin Bayes-verkon rakenne tutkimusaineistossa. Solmu C on lajiluokka. Symboli P merkitsee osaa ja symboli X piirrevektoria. Piirrevektoreiden $X2,\ldots,X8$ yksittäiset solmut on jätetty piirtämättä.

Koska tässä tutkimuksessa luokkien määrä on viisi ja kaikki luokat ovat yhtä todennäköisiä, on kunkin luokan prioritodennäköisyys $P(c_i) = 0,2$. Luokkakohtaiset piirrevektorien uskottavuudet $P(x_1,\cdots,x_{18}|c_i)$ määritettiin muodostamalla koulutusdatan piirrevektoreista tasaväliset histogrammit, joissa kussakin oli 20 lokeroa, ja normalisoimalla näin saadut jakaumat todennäköisyysjakaumiksi. Käytännön toteutuksen kannalta sekä koulutus- että testidatan sisältävä aineisto kannatti ensin diskretisoida luokkiin ja vasta tämän jälkeen muodostaa koulutusdatasta lajikohtaiset histogrammit sekä todennäköisyysjakaumat. Todennäköisyyksien muodostaminen vastasi siten tilannetta, jossa analysoitava data noudattaa multinomiaalijakaumaa ja kukin muuttuja voi saada arvoja joukosta $1,\ldots,20$ (ks. luku 3.3.1). Muuttujan x_i luokan k todennäköisyys $\hat{\theta}_{ijk}$ vanhempiluokan j sisällä saadaan siten kaavalla 3.3. Parametrin θ_{ijk} priorin oletettiin olevan tasainen, joten jokaiseen histogrammiluokkaan lisättiin ykkönen. Tasainen priori takasi täten myös sen, että minkään luokan todennäköisyys ei ollut nolla.

Histogrammien minimi- ja maksimiarvo laskettiin globaalista, kaikki kukat sisältävästä datasta. Näin varmistettiin, että jaettaessa data luvussa 4.3.3 kuvatulla tavalla koulutus- ja testiaineistoon, mallin arvoalue kattaa kaikki testiaineistossa mahdollisesti esiintyvät arvot. Ennen minimin ja maksimin määräämistä data tarkastettiin ulkopuolisten havaintojen varalta. Ulkopuoliset havainnot määritettiin Tukeyn menetelmällä, joka kvartiileihin perustuvana ei riipu jakauman muodosta (Tukey 1977). Menetelmässä poikkeaviksi tulkitaan havainnot, jotka ovat pienempiä kuin $Q1 - 1,5QR$ tai suurempia kuin $Q3 + 1,5QR$. $Q1$ on 25 %:n kvartiili, $Q3$ on 75 %:n kvartiili ja QR näiden välinen erotus. Histogrammin minimiarvoksi valittiin suurempi luvuista datan minimiarvo ja poikkeavien havaintojen alaraja. Maksimiarvoksi puolestaan valittiin pienempi luvuista datan maksimiarvo ja poikkeavien havaintojen yläraja. Näin saadun minimin alittavat havaintoarvot luokiteltiin alimpaan mahdolliseen luokkaan ja maksimiarvon ylittävät puolestaan ylimpään mahdolliseen luokkaan. Käsittelemällä poikkeavat arvot tällä tavoin histogrammit saatiin tasaisemmiksi ja vähemmän vinoiksi ja siten paremmin aineiston olennaisia piirteitä kuvaaviksi.

Koska todennäköisyysjakaumat muodostettiin histogrammien pohjalta, muunnettiin alunperin jatkuva data näin ollen diskreetiksi. Analyysiohjelmistosta riippuen dataa olisi voinut käyttää myös jatkuvana. Jatkuva data oletetaan kuitenkin yleensä normaalijakautuneeksi, eivätkä kaikki tämän analyysin piirrevektorit tukeneet tätä oletusta. Edelleen analyysi toteutettiin Pythonin pgmpy-kirjastolla, joka ei analysointihetkellä tukenut jatkuvia muuttujia.

4.3.3 Luokittelutulokset

Naiivin Bayes-luokittelijan keskus- ja reunaosamallin sopivuutta aineistoon tarkasteltiin ristiinvalidointimenetelmällä, jossa joka kierroksella yhdellä koulutusaineiston ulkopuolelle jätetyllä havaintoarvolla testataan muiden havaintoarvojen perusteella rakennettua mallia (Arlot ja Celisse 2010). Validointikierrosten määrä vastaa näin ollen havaintojen määrää. Menetelmää kutsutaan nimellä *leave one out*, ja siihen päädyttiin, koska havaintoaineiston määrä oli suhteellisen pieni, 400 kuvaa. Testidatan luokan ennustearvo laskettiin viestinvälitysalgoritmilla[8] (ks. luku 3.2.2).

8. Ks. pgmpy-dokumentaatio: inference.ExactInference

Taulukossa 1 on kuvattu mallin sekaannusmatriisi[9] eli kunkin lajin osalta osuudet lajiluokista, joihin testikuvan on ennustettu kuuluvan. Ennustetut lajiluokat sijaitsevat sarakkeilla, joten riviprosentit summautuvat pyöristystarkkuuden rajoissa sataan. Luokittelutarkkuudet, eli osuudet oikein osuneista ennustuksista, vaihtelevat 71,3 ja 97,5 %:n välillä. Orvokkien luokittelu onnistuu parhaiten, mikä oli odotettavissa, sillä laji eroaa muista niin väreiltään kuin yleismuodoltaan. Hankalimmin toisistaan erotettavissa ovat voikukka ja niittyleinikki, jotka molemmat ovat keltaisia, pyöreähköjä kukkia ilman selkeästi erottuvaa keskusosaa. Valkoterälehtiset ja keltakeskustaiset valkovuokot ja päivänkakkarat sotkeutuvat niinikään ajoittain toisiinsa.

Taulukko 1: Naiivin Bayes-menetelmän sekaannusmatriisi (%) keskus- ja reunaosamallissa.

	Päivänkakkara	Voikukka	Leinikki	Valkovuokko	Orvokki
Päivänkakkara	81,3	0,0	1,3	13,8	3,8
Voikukka	0,0	77,5	20,0	1,3	1,3
Leinikki	1,3	26,3	71,3	0,0	1,3
Valkovuokko	10,0	0,0	0,0	88,8	1,3
Orvokki	1,3	1,3	0,0	0,0	97,5

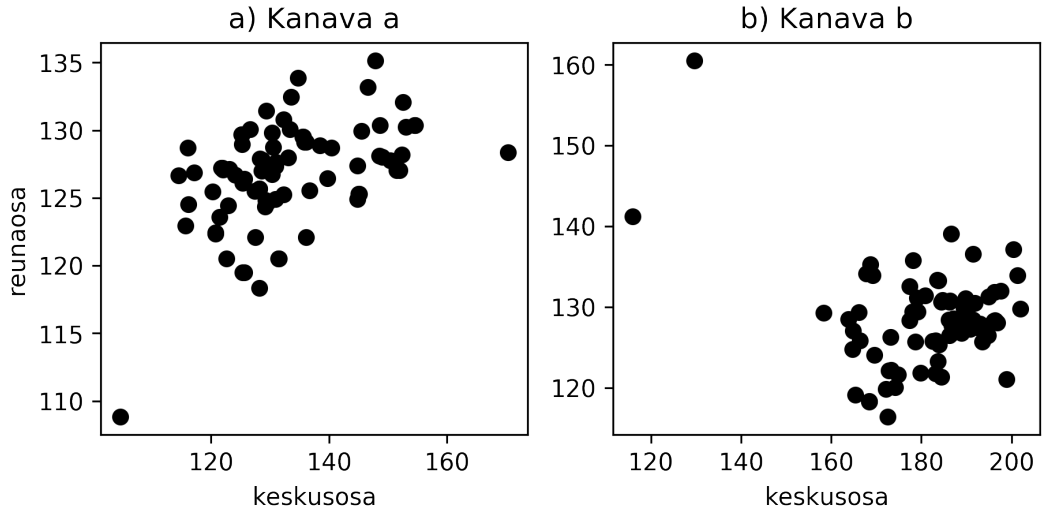
4.3.4 Havaintoja menetelmän ominaisuuksista ja käyttökelpoisuudesta

Naiivi Bayes-menetelmä luokitteli suurimman osan kasveista oikeaan luokkaan. On syytä kuitenkin muistaa, että kohteet oli valmiiksi ikkunoitu, kuvassa oli pääsääntöisesti vain yksi kohde ja lajeja oli vain viisi.

Kohteen määrittäminen ja eritoten osiin segmentointi voi olla ongelmallista myös kohteen reunoihin tiiviisti rajatuissa kuvissa. Erityisesti orvokin tummat osat jäivät helposti kohteen ulkopuolelle (kuviot 9 ja 10). Kukan keskusta on hankala määrittää, jos kukka on ylivalottunut tai kuvattu toisesta kuvakulmasta (kuvio 12). Tällöin piirrevektorit lasketaan väärin määritellyistä segmenteistä ja ne poikkeavat muista saman lajin yksilöistä vaikuttaen heikentävästi tunnistustarkkuuteen (kuvio 13).

9. Ks. sklearn-dokumentaatio: confusion_matrix

Kuvio 12: Väärin segmentoidut kukat.



Kuvio 13: Päivänkakkaroiden keskus- ja reunaosan Lab-väriavaruuden värikanavien keskiarvot.

Kohteen keskipiste määritettiin kynnystetyn kuvan massakeskipisteenä. Vääränlainen kuvan segmentointi vaikuttaa siten myös keskipisteen sijaintiin eikä keskipiste siksi aina välttämättä sijaitse kukan keskustassa, kuten olisi tarkoitus. Lisäksi joidenkin lajien, kuten orvokin, kukat eivät ole ympyrämäisesti symmetrisiä, jolloin oikeampi keskipisteen sijainti olisikin hieman sivussa varsinaisesta massakeskipisteen sijainnista. Keskipisteenä voisikin olla perusteltua käyttää myös käyttäjän määrittelemää kukan keskustaa, jos ajatellaan, että kyse olisi esimerkiksi kasveja tunnistavasta älypuhelinsovelluksesta.

Segmentoiduissa kuvissa esiintyi myös jonkin verran osiin luokiteltuja varsinaisen kohteen ulkopuolisia segmenttejä, jotka vaikuttavat jossain määrin momenttiarvoihin. Lisäksi suurimmalla osalla lajeista eritoten keskiosan muoto on samantyyppinen, pyöreähkö, ja lajien sisäinen variaatio suurta, jolloin osan yleismuotoa kuvaavat momentit eivät kykene erottelemaan lajeja riittävästi. Hienojakoisempi osajako esimerkiksi yksittäisten terälehtien erottele-

miseksi voisi olla tarpeen. Kohteen muotoa tutkittaessa olisi voinut tarkastella myös kohteen reunakäyrää ja sen vaihtelujen taajuutta, mikä tosin edellyttäisi katkeamatonta ja yksikäsitteistä reunakäyrää. Esimerkiksi valkovuokon pistemäisistä segmenteistä koostuvalle keskusosalle olisi hankala määrittää reunakäyrää (kuvio 10). Väärä segmentointi aikaansaisi myös väärän reunakäyrän.

Myös Nilsback ja Zisserman (2006) kokivat kukkien segmentoinnin haasteellisena, minkä vuoksi he päätyivät parantamaan artikkelinsa segmentointimenetelmää kukan muotomallilla (Nilsback ja Zisserman 2007). Alkuperäisessä segmentointimenetelmässä jokaisesta lajiluokasta poimittiin muutama kuva, joissa osa pikseleistä määritettiin manuaalisesti kohteeseen ja osa taustaan kuuluviksi (Nilsback ja Zisserman 2006). Näistä näytepikseleistä saadut kohteen ja taustan värijakaumat keskiarvoistettiin koskemaan koko aineistoa. Kuvien loput pikselit luokiteltiin automaattisesti vertaamalla kohdepikselin väriä näytejakaumiin sekä naapuripikseleiden väriarvoihin ja etsimällä näin rajat, jotka optimaalisimmin erottelivat pikselit kuuluvaksi joko kohteeseen tai taustaan (Boykov ja Jolly 2001; Nilsback ja Zisserman 2006). Menetelmä ei kuitenkaan kaikissa tapauksissa tuottanut täysin onnistunutta segmentointia: esimerkiksi orvokin tummat osat saattoivat jäädä kohteen ulkopuolelle (Nilsback ja Zisserman 2006). Nilsback ja Zisserman (2007) kehittivät edellä kuvattua menetelmää seuraavasti. Kuva segmentoitiin ensin värijakaumamenetelmällä, minkä jälkeen kohteeseen sovitettiin kuvakohtainen kukan muotomalli, joka tunnisti potentiaaliset terälehdet. Uuden kohteen muodosti se osa vanhasta kohteesta, johon kukan muotomalli pystyttiin sovittamaan. Kohteen värijakauma laskettiin uudesta kohteesta ja kuva segmentoitiin uudestaan kohteen ja taustan värijakaumiin perustuvalla menetelmällä. Prosessia jatkettiin, kunnes segmentointi ei enää muuttunut. Segmentointi parani selkeästi alkuperäisestä (Nilsback ja Zisserman 2007). Myös luokittelutulokset paranivat keskimäärin prosenttiyksikön verran (Nilsback ja Zisserman 2008).

Yhteenvetona taulukon 1 tuloksista havaitaan, että eniten toisiinsa sekaantuivat samanväriset kukat, joten luokittelu nojasi vahvasti kuvasta saatuun väri-informaatioon. Myös kukan segmentointi keskus- ja reunaosaan tapahtui suurelta osin värin perusteella. Lajien tunnistustarkkuuden parantamiseksi tarvitaankin piirteitä, jotka kuvaavat kukan muotoa yksityiskohtaisemmin.

4.4 Naiivi Bayes-luokittelija ja muotopiirteet

Luvussa 4.3.4 todettiin naiivin Bayes-mallin toimivan luokittelijana varsin hyvin, mutta mallin piirteiden nojaavan liikaa kuvien väri-informaatioon. Ennen kuin siirryttiin piirremuuttujien riippuvuuksia huomioiviin malleihin, pyrittiin siksi ensin määrittämään, millaisilla piirteillä kukkien muotoa voitaisiin kuvata ensimmäistä mallia paremmin.

4.4.1 Muotoa kuvaavien piirrevektoreiden muodostaminen

Suosittuja ja kohtuullisen hyvin toimivia kohteen muotoa kuvaavia piirteytysmenetelmiä ovat skaalainvariantit pistepiirteet (ks. luku 2.3.1), joita on käytetty usein myös kukkien tunnistamisessa (Wäldchen ja Mäder 2017). Yhtenä tämän tutkimuksen tavoitteena oli pistepiirteitä hyödyntämällä yrittää löytää kukasta terälehtien kärjet ja niiden myötä itse terälehdet, joiden muodon ja sijainnin toisiinsa nähden voidaan ajatella olevan kullekin lajille ominaisia.

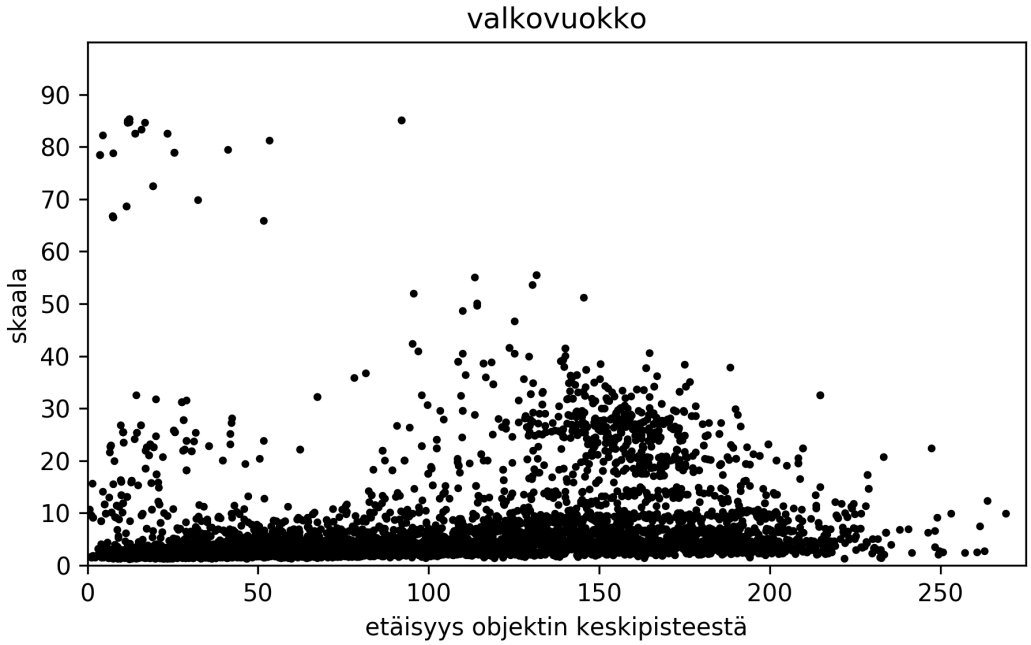
Skaalainvarianteista pistepiirteistä kokeiltiin sekä SIFT- että SURF-menetelmää[10] (Lowe 2004; Bay ym. 2008). SIFT-menetelmän löytämät avainpisteet osuivat silmämääräisesti tarkasteltuna SURF-menetelmän pisteitä paremmin terälehtien kärkiin, joten menetelmäksi valittiin SIFT. Menetelmä on kuvattu pääpiirteittäin luvussa 2.3.4. Vaikutti kuitenkin siltä, että OpenCV:n SIFT-toteutus ei löytänyt kaikkia sellaisia terälehtiä, jotka sen silmämääräisesti arvioituna olisi tullut löytää. Tämän vuoksi SIFT-algoritmin avainpisteiden etsimisosuus, mukaanlukien lokaalien maksimien etsintä ja matalakontrastisten ja reunapisteiden poisto, päädyttiin toteuttamaan myös itse. Pythonilla itse toteutettu SIFT-algoritmin alkuosa löysikin terälehtien kärkiin sijoittuvia avainpisteitä OpenCV:n vastaavaa toteutusta paremmin. Koska pro gradu -työn keskiössä ei ollut pistepiirrealgoritmien rakentaminen ja koko SIFT-algoritmin toteuttaminen olisi ollut suhteellisen työläs operaatio, päädyttiin lopulta käyttämään Vedaldin ja Fulkersonin (2008) Matlabilla koodaamaa SIFT-toteutusta[11], joka kuvista arvioituna löysi jotakuinkin samat avainpisteet kuin itse toteutettu SIFT. OpenCV:n SIFT-algoritmin koodia katselmoimalla ei selvinnyt, miksi se löysi muita toteutuksia huonommin terälehtien kärkiä. Kuvan rajaus kohteeseen oli melko tiivis, mikä on saattanut vaikuttaa avainpisteiden löytymiseen.

10. Ks. OpenCV 2.4 -dokumentaatio: SIFT ja SURF
11. Ks. VLFeat-dokumentaatio: SIFT

Avainpisteitä etsittäessä parametrien arvoja muunneltiin jonkin verran. Eniten vaikutusta oli kontrastin raja-arvolla, joka päädyttiin laskemaan arvoon $0,025$ Lowen (2004) suosittelemasta arvosta $0,03$. Tämän voi perustella kuvien epäterävyydellä. Kontrastin raja-arvoa pienentämällä saatiin erityisesti kukan keskusosasta lisää avainpisteitä ja siten informaatiota. Gaussisen suodatuksen alkuarvoksi valittiin Lowen suosittelema $\sigma = 1,6$. Myös reunan ja nurkan erotteluparametrin raja-arvoksi valittiin oletusarvo $r = 10$. Skaalojen määräksi valittiin kaksi, koska se tuntui silmämääräisesti arvioiden löytävän hieman paremmin terälehtien kärkiä kuin Lowen suosittelema arvo kolme. Suurta eroa ei kuitenkaan ollut havaittavissa.

Avainpisteet määritettiin normalisoidusta harmaasävykuvasta. Ne määritettiin koko kuvan alueelta, mutta analyyseihin valikoitiin vain maskin sisään jääneet avainpisteet. Maski muodostettiin yhdistämällä ensimmäisessä mallissa kuvattu keskus- ja reunaosa yhtenäiseksi alueeksi ja paikkaamalla näin saadun alueen sisään jäävät reiät.

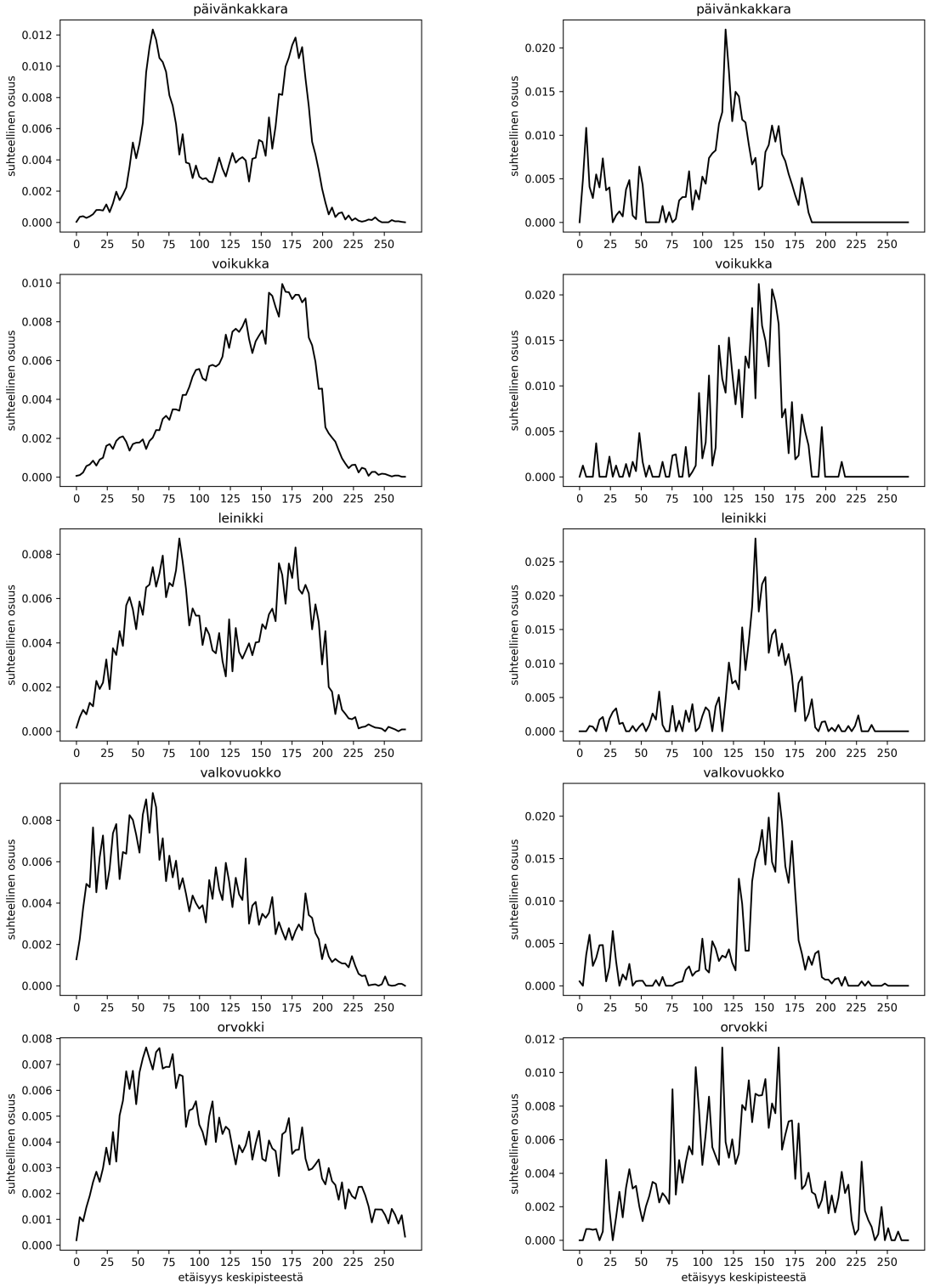
Terälehtien kärjissä sijaitsevat avainpisteet pyrittiin löytämään tutkimalla avainpisteiden geometrisia ominaisuuksia. Terälehtien koko vaihtelee lajeittain. Täten voidaan olettaa, että tietyllä lajilla terälehdet löytyvät tietystä avainpisteiden skaalaluokasta edellyttäen, että kuvat on skaalattu samaan kokoon, kuten tässä tapauksessa oli asian laita. Toisin sanoen voidaan ajatella, että tietyn kokoluokan avainpisteet kuvaavat tiettyjä kukan ominaisuuksia. Toisaalta kukassa on yleensä havaittavissa keskus- ja reunaosa. Tällöin tietyn etäisyysluokan avainpisteiden voidaan jälleen ajatella kuvaavan tiettyjä kukan ominaisuuksia, kun luokan etäisyys on mitattu kohteen keskipisteestä. Esimerkiksi terälehtien voi olettaa löytyvän reunaosasta. Skaala- ja etäisyysluokkien määrittämiseksi avainpisteet kuvattiin etäisyys (kohteen keskipisteestä) ja skaala (avainpisteen koko) -koordinaatistossa lajeittain. Esimerkkinä tästä on valkovuokon pisteiden jakautuminen (kuvio 14); kaikki lajikuviot löytyvät liitteestä A. Etäisyys vs. skaala -kuvioista havaitaan, että suurin osa avainpisteistä sijoittuu skaalaluokkaan $]0,15[$. Suurimmalla osalla lajeista skaalaluokan $[55,\infty[$ avainpisteet taas ovat satunnaisia ja niiden voidaankin useimmiten olettaa ilmentävän koko kukan löytymistä. Tällöin saadaan seuraavat skaalaluokat: pienet avainpisteet: $]0,15[$, suuret avainpisteet: $[15,55[$, ja poikkeavan suuret satunnaiset avainpisteet: $[55,\infty[$. Kuviossa 15 on kuvattu kohteen keskipisteestä mitattujen avainpisteiden etäisyyksien jakaumat skaalaluokittain. Jakaumakuvista havaitaan,



Kuvio 14: Valkovuokoista löydetyt avainpisteet skaalan ja kohteen keskipisteestä mitatun etäisyyden suhteen kuvattuna.

että pienen skaalaluokan jakaumissa on kaksihuippuisuutta huippujen välisen pohjan sijoittuessa 100 pikselin tienoille (kuvio 15a). Kun kuvien leveys on 400 pikseliä ja kohde sijoittuu jotakuinkin kuvan keskelle, 100 pikseliä on myös arvo, joka sijaitsee kohteen keskustan ja kuvan reunan puolivälissä. Pienen skaalan keskus- ja reunaosan etäisyysluokkien rajaksi saatiin täten 100 pikseliä. Kuvattaessa suuren skaalan avainpisteiden etäisyydet keskipisteestä jakaumina lajeittain havaitaan, että 100 pikseliä parempi arvo voisikin olla 70 (kuva 15b). Raja-arvolla 100 joitakin potentiaalisia terälehtiä voisi jäädä reunaosaluokan ulkopuolelle. Pienien terälehtien luokan raja-arvot olivat täten seuraavat: skaala $]0,15[$ ja etäisyys $[100,\infty[$. Suurien terälehtien luokan raja-arvot olivat seuraavat: skaala $[15,55[$ ja etäisyys $[70,\infty[$. Kuvioon 16 on piirretty muutaman esimerkkikukan avainpisteet, joiden skaala- ja etäisyysluokat on kuvattu värikoodein. Visuaalisen tarkastelun perusteella jako skaala- ja etäisyysluokkiin vaikuttaa toimivan kohtuullisen hyvin.

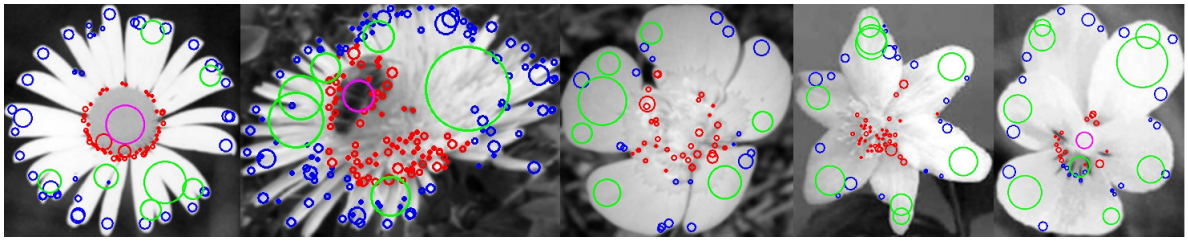
Kuviosta 16 kuitenkin havaitaan, että terälehtiluokkiin sisältyy myös muita avainpisteitä kuin terälehtien kärkiä kuvastavia. Terälehden kärkeen voi osua myös useampi avainpiste. Jotta saataisiin laskettua potentiaalisia terälehtiä kuvaavia piirteitä, ylimääräiset avainpis-



(a) Pienen skaalan pisteet

(b) Suuren skaalan pisteet

Kuvio 15: Kohteen keskipisteestä mitattujen avainpisteiden etäisyyksien jakaumat pienen ja suuren skaalan avainpisteluokissa.

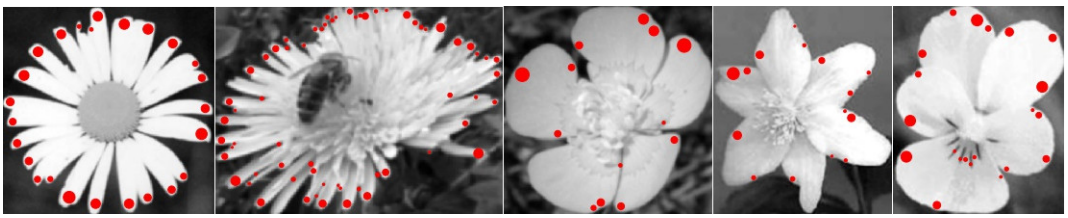


Kuvio 16: SIFT-algoritmin löytämät avainpisteet skaala- ja etäisyysluokissa. Reunaosan pienen skaalaluokan pisteet ovat sinisiä ja ison skaalaluokan pisteet vihreitä. Keskiosan pienen skaalaluokan pisteet ovat punaisia ja ison skaalaluokan pisteet violetteja.

teet tulisi poistaa. Terälehtiä kuvaavat avainpisteet määritettiin kummallekin skaalaluokalle seuraavasti:

1. Laskettiin kunkin avainpisteen etäisyys kohteen keskipisteeseen ja poistettiin pisteet, jotka tulkittiin poikkeaviksi Tukeyn (1977) ulkopuolisia havaintoja löytävällä menetelmällä (ks. luku 4.3.2).
2. Etsittiin kauimpana keskipisteestä oleva piste ja valittiin se terälehtipisteeksi.
3. Poistettiin kaikki ne vielä mukana olevat avainpisteet, jotka jäivät valitun terälehtipisteen ja kohteen keskipisteen muodostaman sektorin sisään. Sektorin leveyden määräsi avainpisteen koko. Avainpisteen tulkittiin jäävän sektorin sisään, jos jokin osa avainpisteeseen piirretystä ympyrästä jäi sektorin sisäpuolelle. Avainpisteympyrän koko vastasi avainpisteen skaalaa.
4. Toistettiin algoritmia kohdasta 2, kunnes avainpisteet loppuivat.

Kuvioihin 17 ja 18 on piirretty isossa ja pienessä skaalassa löytyneet, terälehtiä kuvastavat avainpisteet. Kohteet ovat samoja kuin kuviossa 16, joihin on piirretty avainpisteet värikoodatuissa skaala- ja etäisyysluokissa. Vertaamalla sinisiä avainpisteitä kuvion 17 pisteisiin ja vihreitä avainpisteitä kuvion 18 pisteisiin voidaan tarkastella algoritmin toimivuutta.



Kuvio 17: Avainpisteet, jotka kuvastavat pieniä terälehtiä.

Kuvio 18: Avainpisteet, jotka kuvastavat isoja terälehtiä.

Terälehtien keskinäistä sijaintia kuvaavien piirteiden laskemiseksi määritettiin terälehtipisteen ja ylöspäin osoittavan y-akselin välinen kulma ja terälehtipisteet järjestettiin kulman perusteella. Origona kulman laskemisessa toimi kohteen keskipiste. Kullekin kuvalle ja terälehtiluokalle laskettiin seuraavat piirteet:

- kulman suhteen peräkkäisten pisteiden välisen etäisyyden keskiarvo,
- kulman suhteen peräkkäisten pisteiden välisen etäisyyden keskihajonta,
- kulman suhteen peräkkäisten pisteiden välisen kulman keskiarvo,
- kulman suhteen peräkkäisten pisteiden välisen kulman keskihajonta.

Jos lajin terälehdet ovat pieniä ja niitä on paljon, voidaan olettaa, että pienien terälehtien luokassa esiintyy suurien terälehtien luokkaa enemmän säännönmukaisuutta terälehtipisteiden sijainnissa. Jos taas lajin terälehdet ovat isoja ja niitä on vain muutamia, säännönmukaisuutta voidaan olettaa löytyvän ennen kaikkea suurien terälehtien luokassa. Ongelmana toki on, että pieniä avainpisteitä löytyy suuria enemmän, mikä on jo sinällään säännönmukaisuutta lisäävä tekijä.

Avainpisteiden sijaintia piirteistettiin myös laskemalla niiden sijaintia kuvaavat Hun momentit (Hu 1962). Momentit laskettiin erikseen pienen skaalan keskus- ja reunaluokissa; suuren skaalan avainpisteitä oli sen verran vähän, että momentit laskettiin koko luokan pisteiden perusteella. Pienen skaalan pisteillä yleensä selvästi runsaslukuisemmat reunaosan pisteet olisivat saattaneet peittää alleen keskusosan pisteiden momenttien mahdollisia eroja. Esimerkiksi kuviossa 16 päivänkakkaran (1. kukka) ja valkovuokon (4. kukka) keskusosan pienet avainpisteet sijoittuvat toisistaan poikkeavasti, minkä voisi olettaa näkyvän momenttien arvoissa. Momenttien laskennassa keskipisteenä toimi kukan keskipiste ja kynnysmaskina avainpisteisiin piirretyt, pisteen skaalaa kuvastavat ympyrät. Momenteista päädyttiin valitse-

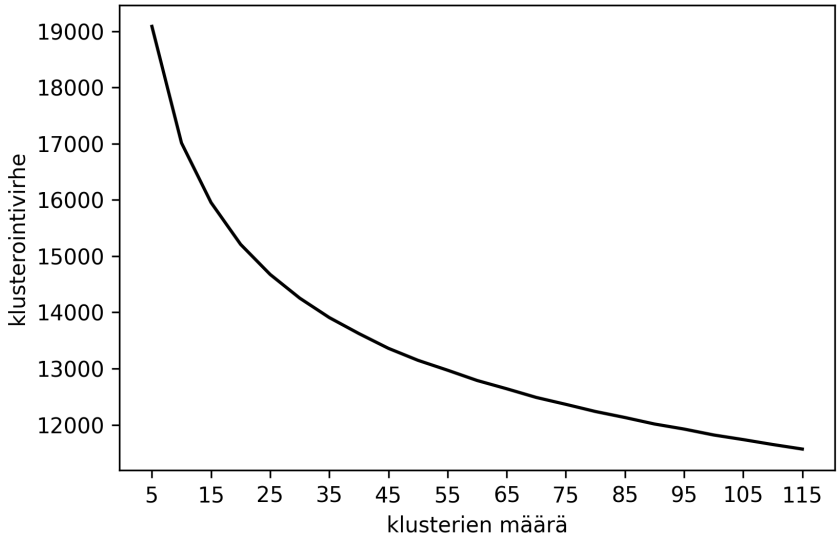
maan piirteeksi vain Hun 1. momentti, koska logaritmiarvoiksi muunnettuina sen hajonnat yksilöiden sisällä olivat kohtuullisia ja jakaumat suhteellisen siistejä muihin momentteihin verrattuna. Hun 1. momentille on olemassa myös selkeä tulkinta hitausmomenttina (esim. Donchenko ja Golik 2013), joka tässä tapauksessa kuvastaa, millä etäisyydellä tietyn luokan avainpisteet sijaitsevat kohteen keskipisteestä, ja kuinka paljon ja minkäkokoisina pisteitä löytyy. Muut Hun momentit ovat hankalasti tulkittavissa.

Varsinaisiin skaalainvariantteihin pistepiirteisiin eli SIFT-algoritmin tuottamiin deskriptoreihin valittiin kerralla kaikki maskin kattamien avainpisteiden deskriptorit; siten deskriptoreihin ei enää sovellettu luokkajakoa, vaikka niinkin olisi voinut tehdä. Koska SIFT-algoritmin tuottamien deskriptorien dimensio on varsin korkea, 128, ja joidenkin deskriptorimuuttujien havaittiin korreloivan, tehtiin deskriptoreille alkukäsittelynä pääkomponenttianalyysi[12] (Zaki ja Meira 2014, luku 7.2). PCA:ta varten data keskiarvoistettiin nollan ympärille. Pääkomponenttien määräksi valittiin 53 komponenttia, jotka kattoivat yli 90 % havaitusta vaihtelusta. Seuraavaksi pääkomponenteista muodostettiin uudet muuttujat *bag-of-features* -menetelmällä (ks. luku 2.3.5). Deskriptorit luokiteltiin ensin ryhmiin k-means-klusterointimenetelmällä[13] (Zaki ja Meira 2014, luku 13.1). Kuvioiden 19 ja 20 perusteella sopivaksi klustereiden määräksi valittiin 20. Kuviossa 19 on kuvattu klustereiden sisäiset neliösummat klustereiden määrän funktiona. Klustereiden määrä valittiin niin sanotulla *elbow*-metodilla, jossa neliösummakuvasta etsitään kohta, jossa käyrä taipuu voimakkaasti (esim. Tibshirani, Walther ja Hastie 2002). Lisäksi kuviosta 20 havaitaan, että lajeittain kuvattujen klustereiden keskihajontojen keskiarvojen pieneneminen alkaa tasaantua, kun klustereita on 20 tai enemmän.

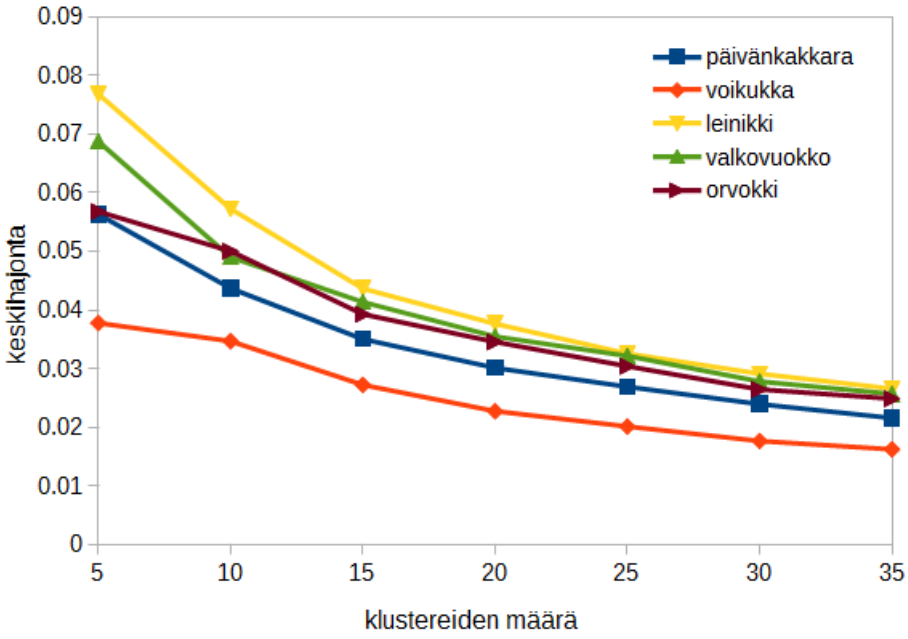
Klusteroinnin jälkeen kullekin yksilölle laskettiin, montako deskriptoria mihinkin ryhmään oli luokiteltu. Tällä tavoin saatu jakauma normalisoitiin, jotta saatiin vakioitua pois löytyneiden avainpisteiden määrän vaikutus. Normalisoidut arvot muodostivat uudet *bag-of-features* eli *bof*-muuttujat. Koska muuttujia oli edelleen melko paljon, 20, ja niissä havaittiin korrelaatiota, joka kuvien perusteella oli jotakuinkin lineaarista, päädyttiin myös *bof*-muuttujille tekemään pääkomponenttianalyysi. Kuvion 21 perusteella, jossa on kuvattu kunkin kompo-

12. Ks. matplotlib-dokumentaatio: PCA
13. Ks. sklearn-dokumentaatio: KMeans

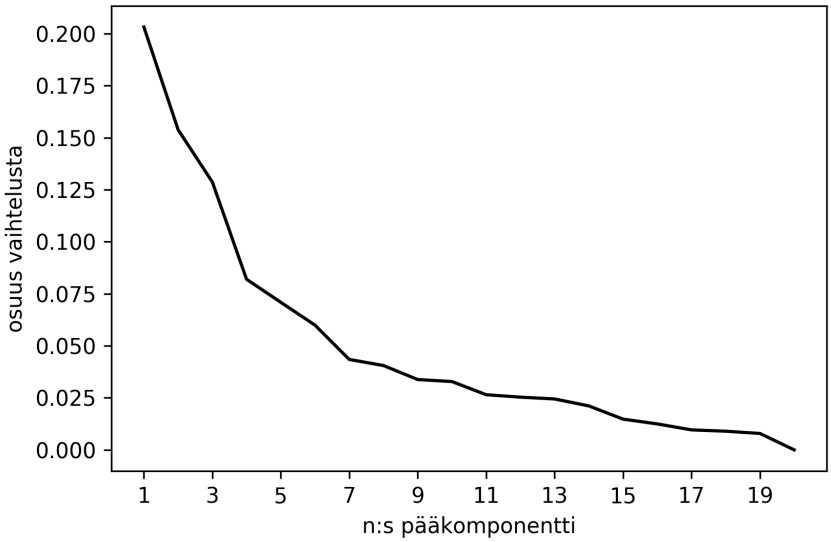
nentin sisältämä osuus kokonaisvaihtelusta, pääkomponentteja valittiin 7, ja ne kattoivat yli 75 % kokonaisvaihtelusta.



Kuvio 19: Klusteroidun *bag-of-features* -datan virheneliösumma.



Kuvio 20: Klusteroidun *bag-of-features* -datan klustereittain laskettujen hajontojen keskiarvot lajiluokissa.



Kuvio 21: *Bag-of-features* -datan pääkomponenttien osuudet kokonaisvaihtelusta.

4.4.2 Mallin kuvaus ja piirrevektorit

Mallina oli ensimmäisestä mallista tuttu naiivi Bayes-malli, jota täydennettiin luvussa 4.4.1 kuvatuilla muotopiirteillä. Näin ollen malli sisälsi seuraavat 36 piirrevektoria:

- keskus- ja reunaosan Lab-väriavaruuden a- ja b-kanavan keskiarvo, keskihajonta sekä jakaumien vinousarvo,
- keskus- ja reunaosan Hun 2. ja 4. kuvamomentti (Hu 1962),
- osien keskimääräiset suhteelliset euklidiset etäisyydet kohteen keskipisteestä,
- kulman suhteen peräkkäisten pisteiden välisen etäisyyden keskiarvo pienessä ja isossa terälehtiluokassa,
- kulman suhteen peräkkäisten pisteiden välisen etäisyyden keskihajonta pienessä ja isossa terälehtiluokassa,
- kulman suhteen peräkkäisten pisteiden välisen kulman keskiarvo pienessä ja isossa terälehtiluokassa,
- kulman suhteen peräkkäisten pisteiden välisen kulman keskihajonta pienessä ja isossa terälehtiluokassa,

- Hun 1. momentti laskettuna ison skaalan avainpisteille ja erikseen pienen skaalan keskimmäisille ja reunimmaisille avainpisteille,
- 7 deskriptori-muuttujaa.

Piirrevektorit on nimetty niihin viittaamisen helpottamiseksi. Nimilyhenteet on luetteloitu taulukoissa 2, 3 ja 4.

Taulukko 2: Kukan väriä kuvaavien piirremuuttujien lyhenteet.

Lyhenne	Selite
cent:labA_mean	Keskusosan Lab-väriavaruuden a-kanavan keskiarvo.
cent:labA_sd	Keskusosan Lab-väriavaruuden a-kanavan keskihajonta.
cent:labA_skew	Keskusosan Lab-väriavaruuden a-kanavan vinousarvo.
cent:labB_mean	Keskusosan Lab-väriavaruuden b-kanavan keskiarvo.
cent:labB_sd	Keskusosan Lab-väriavaruuden b-kanavan keskihajonta.
cent:labB_skew	Keskusosan Lab-väriavaruuden b-kanavan vinousarvo.
edge:labA_mean	Reunaosan Lab-väriavaruuden a-kanavan keskiarvo.
edge:labA_sd	Reunaosan Lab-väriavaruuden a-kanavan keskihajonta.
edge:labA_skew	Reunaosan Lab-väriavaruuden a-kanavan vinousarvo.
edge:labB_mean	Reunaosan Lab-väriavaruuden b-kanavan keskiarvo.
edge:labB_sd	Reunaosan Lab-väriavaruuden b-kanavan keskihajonta.
edge:labB_skew	Reunaosan Lab-väriavaruuden b-kanavan vinousarvo.

Naiivin Bayes-luokittelijan rakentaminen on kuvattu luvussa 4.3.2. Erona ensimmäiseen naiiviin Bayes-malliin tämän mallin joistakin piirremuuttujista puuttui havaintoja. Kaikilta yksilöiltä ei esimerkiksi havaittu ison skaalan reunaosan avainpisteitä. Tällaisten piirremuuttujien jakaumat laskettiin samoin kuin kaikki havainnot sisältävien piirremuuttujien jättämällä jakaumaa muodostettaessa yksinkertaisesti pois laskuista yksilö, jolta havainto puuttui. Ennustettaessa yksilön luokkaa käytettiin kaikkia saatavilla olevia havaintoja. Siten joidenkin yksilöiden lajiluokkaa ennustettiin kaikilla muuttujilla, kun taas toisten yksilöiden lajiluokkaa ennustettaessa joitakin muuttujia jouduttiin jättämään pois.

Taulukko 3: Terälehtien sijaintia kuvaavien piirremuuttujien lyhenteet.

Lyhenne	Selite
S:mean_dist	Pisteiden välisen etäisyyden keskiarvo pienessä terälehtiluokassa.
S:sd_dist	Pisteiden välisen etäisyyden keskihajonta pienessä terälehtiluokassa.
S:mean_angle	Pisteiden välisen kulman keskiarvo pienessä terälehtiluokassa.
S:sd_angle	Pisteiden välisen kulman keskihajonta pienessä terälehtiluokassa.
L:mean_dist	Pisteiden välisen etäisyyden keskiarvo isossa terälehtiluokassa.
L:sd_dist	Pisteiden välisen etäisyyden keskihajonta isossa terälehtiluokassa.
L:mean_angle	Pisteiden välisen kulman keskiarvo isossa terälehtiluokassa.
L:sd_angle	Pisteiden välisen kulman keskihajonta pienessä terälehtiluokassa.

4.4.3 Luokittelutulokset

Kukan muotopiirteillä vahvistetun mallin sopivuutta aineistoon tarkasteltiin jälleen *leave one out* -ristiinvalidoinnilla. Taulukossa 5 on esitetty muotopiirteillä vahvistetun naiivin mallin sekaannusmatriisi. Vertailtaessa ensimmäisen mallin tuloksia (taulukko 1) uuden mallin tuloksiin (taulukko 5) luokittelutarkkuuden havaitaan parantuneen kaikkien muiden lajien paitsi orvokin kohdalla — joskin orvokeista alunperinkin vain kaksi kuvaa luokittui väärään luokkaan. Muotopiirteiden lisäys malliin vaikutti erityisesti voikukkaan ja leinikkiin, joiden luokittelutarkkuudet paranivat 20 prosenttiyksikköä.

4.4.4 Havaintoja menetelmän ominaisuuksista ja käyttökelpoisuudesta

Koska luokittelutarkkuus parani huomattavasti ensimmäiseen malliin verrattuna erityisesti voikukan ja leinikin osalta, malliin valittujen muotopiirteiden voidaan tulkita sisältäneen sellaisia kukan muoto-ominaisuuksia, jotka toivat kaivattua lisäinformaatiota luokitteluun. Muotopiirteitä voidaan kuitenkin muodostaa monella tavalla, joten varmasti ei voida sanoa valittujen piirteiden olleen parhaat mahdolliset.

SIFT on tunnettu ja myös kasvilajien tunnistuksessa yleisesti käytetty algoritmi, jonka tuottamat piirteet ovat skaala- ja rotaatioinvariantteja (Lowe 2004; Wäldchen ja Mäder 2017). Al-

Taulukko 4: Kukan muotoa sekä reuna- ja keskiosan kokoa kuvaavien piirremuuttujien lyhenteet.

Lyhenne	Selite
cent:dist	Keskusosan suhteellinen etäisyys kohteen keskipisteestä.
edge:dist	Reunaosan suhteellinen etäisyys kohteen keskipisteestä.
cent:hu2	Keskusosan Hun 2. kuvamomentti.
cent:hu4	Keskusosan Hun 4. kuvamomentti.
edge:hu2	Reunasosan Hun 2. kuvamomentti.
edge:hu4	Reunasosan Hun 4. kuvamomentti.
S1:hu1	Pienen skaalan keskiosan avainpisteiden Hun 1. momentti.
S2:hu1	Pienen skaalan reunaosan avainpisteiden Hun 1. momentti.
L:hu1	Ison skaalan avainpisteiden Hun 1. momentti.
d1	SIFT-deskriptori 1.
d2	SIFT-deskriptori 2.
d3	SIFT-deskriptori 3.
d4	SIFT-deskriptori 4.
d5	SIFT-deskriptori 5.
d6	SIFT-deskriptori 6.
d7	SIFT-deskriptori 7.

goritmi tuottaa kuitenkin runsaasti 128-alkioisia piirrevektoreita kuvaa kohti, joten yleinen tapa on edelleenkäsitellä niitä esimerkiksi *bag-of-features* -menetelmällä (Zhang ym. 2013). Tässä tutkimuksessa deskriptoreille tehtiin myös pääkomponenttianalyysi ennen kuin data käsiteltiin *bof*-menetelmällä. Myös *bof*-menetelmän tuottamat piirteet käsiteltiin pääkomponenttianalyysilla. Ensimmäisessä PCA-käsittelyssä komponenttien määrä valittiin kattamaan 90 % kokonaisvaihtelusta, mikä lienee riittävä määrä aineiston luotettavaan kuvaukseen. Jälkimmäisessä käsittelyssä komponenttien lukumäärä valittiin kuvan perusteella, jossa kuvattiin vaihtelun osuus komponenteittain. Komponentteja valittiin seitsemän, mutta itse asiassa komponenttien määräksi olisi todennäköisesti riittänyt kuusi, sillä komponenttien 7 ja 8 välillä ei ole enää suurta eroa niiden kattamassa vaihtelussa. Todennäköisesti komponent-

Taulukko 5: Muotopiirteillä vahvistetun naiivin Bayes-mallin sekaannusmatriisi (%).

	Päivänkakkara	Voikukka	Leinikki	Valkovuokko	Orvokki
Päivänkakkara	88,8	0,0	1,3	6,3	3,8
Voikukka	0,0	97,5	1,3	0,0	1,3
Leinikki	1,3	7,5	91,3	0,0	0,0
Valkovuokko	5,0	0,0	0,0	93,8	1,3
Orvokki	0,0	0,0	0,0	2,5	97,5

teja on valittu molemmissa analyyseissa liikaa. Jackson (1993) on vertaillut tutkimuksessaan erilaisia menetelmiä komponenttien lukumäärän määräämiseksi. Valittaessa komponentteja kokonaisvaihtelun perusteella käytetään usein 95 %:n raja-arvoa. Jackson (1993) kuitenkin toteaa menetelmän yliestimoivan komponenttien tarpeen. Kun määrä valitaan kuvan perusteella, joka sisältää yksittäisen komponenttien kattamat vaihtelut, Jacksonin (1993) mukaan myös valittua arvoa 7 yhtä pienempi olisi yliestimoinut komponenttien määrän heikkojen korrelaatioiden tilanteessa. Pääkomponenttien tapaan myös klustereiden määrän valinta on enemmän tai vähemmän intuition varassa datan ominaisuuksista riippuen. Esimerkiksi kuviossa 19 ei tapahdu selkeää notkahdusta klusterointivirheen vähenemisessä minkään klusterimäärän kohdalla, joten klustereiden määrä olisi hyvin voinut olla myös 25 tai 30. Kuvion 20 perusteella lukumäärä 20 vaikuttaisi kuitenkin riittävältä; pisteet näyttävät x-akselin arvosta 20 lähtien asettuvan melko lailla samalle suoralle. Luonnollisesti myös SIFT-algoritmin parametrien arvot vaikuttavat tuloksiin ja siihen, kuinka paljon ja mistä avainpisteitä löytyy.

Avainpisteiden sijainnin ja koon pohjalta laskettu Hun 1. momentti kuvaa, miten avainpisteet ovat sijoittuneet suhteessa kohteen keskipisteeseen. Esimerkiksi päivänkakkaralla pisteet muodostavat kehää keltaisen teriön ja valkoisten terälehtien väliselle reuna-alueelle. Valkovuokolla avainpisteet sitä vastoin jakautuvat kukan keskusalueella tasaisemmin löytäen yksittäisiä heteitä ja emejä. Koska momenttien pohjadatana käytettiin avainpisteitä, SIFT-algoritmin parametriarvot vaikuttavat tuloksiin. Kuvien terävyys taas vaikuttaa avainpisteiden löytymiseen.

Orvokeilla voidaan havaita kukan keskustan ympärillä tummia mesiviittoja, jotka ohjaavat pölyttäjiä kohti kukan mesivarantoja (Hansen, Van der Niet ja Johnson 2012). Muilta tarkasteltavilta lajeilta ihmissilmin nähtävät mesiviitat puuttuvat, joten kyseessä voisi olla lajeja erotteleva tekijä. Avainpisteitä sijoittuu jonkin verran mesiviittojen ympärille. Ennen kaikkea avainpisteet löytävät kuitenkin pistemäisiä kohteita, koska selkeät reuna-alueet, joissa havaitaan vain yhdensuuntainen gradientti, on rotaatiovariantteina piirteinä suodatettu pois. Toisaalta kukan asento kuvassa voi vaihdella, joten piirteiden täytyisi olla mahdollisimman invariantteja myös asentomuutoksille. Edelleen kuvan terävyys ja resoluutio vaikuttavat siihen, kuinka yksityiskohtaisia piirteitä voidaan löytää. Avoimeksi kysymykseksi jää, pohjaavatko muotopiirteet liikaa avainpisteisiin.

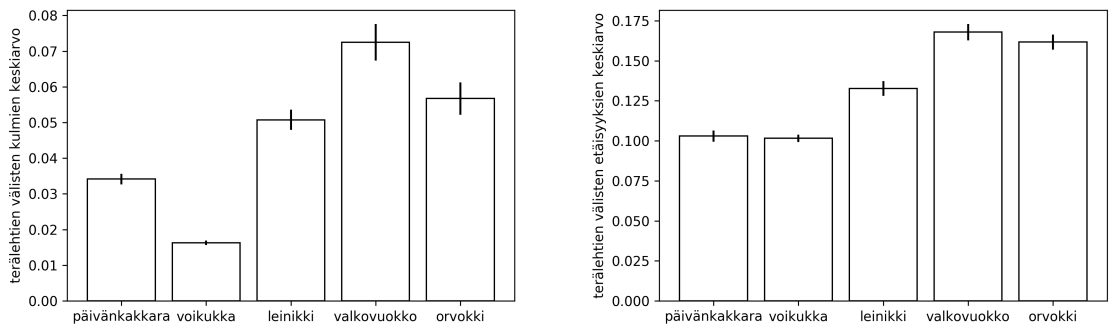
Hun momenteissa ja terälehtiluokissa avainpisteet jaettiin skaalan ja keskipisteestä mitatun etäisyyden mukaan luokkiin. Jako luokkiin on aina jossain määrin keinotekoinen, vaikka luokkarajoja voitaisiinkin perustella esimerkiksi jakaumien kaksihuippuisuudella. Toisaalta sopiva luokkajako voi tuoda esiin oleellista informaatiota. Eräs luokkajaon ongelma on, että skaala ja avainpisteen etäisyys kohteen keskipisteestä määritettiin absoluuttisina arvoina. Menetelmä toimii tässä tapauksessa, jossa kuvat ovat tiiviisti rajatut kohteen reunoihin ja ne ovat samankokoisia. Jos näin ei olisi, luokkajaon tulisi perustua suhteellisiin arvoihin.

Ajatuksena terälehtiluokkien muodostamisessa oli jollakin tapaa kuvata terälehtien sijaintia toisiinsa nähden. Alkuperäisenä tavoitteena oli määrittää kaikille lajeille terälehtien kärjet — ja sitä myötä terälehdet kukan osana — avainpisteiden tai muiden piirteiden avulla, mutta tämä osoittautui hankalaksi. Ratkaisematta esimerkiksi jäi, miten määritellään se, minkäkokoiset avainpisteet kuvaavat kärkiä milläkin lajilla. Täten päädyttiin jakamaan avainpisteet vain isojen ja pienien terälehtien luokkiin.

Taulukkoon 6 on laskettu kahdella eri mittarilla mitattuna, kuinka paljon kukin piirremuuttuja vaikuttaa lajiluokkaan, ja luvut on lajiteltu laskevaan järjestykseen yhteisen informaation mittarin $I_P(A_i;C)$ mukaan. Taulukosta havaitaan, että tiettyjen kukan väriä kuvaavien muuttujien ohella eniten lajiluokkaan ja siten tunnistukseen vaikuttavat pienen terälehtiluokan avainpisteiden välisiä kulmia kuvaavat muuttujat S:mean_angle ja S:sd_angle. Siten ne ovat oleellisessa osassa lajien erottelussa. Kuviosta 22a havaitaan, että päivänkakkaralla, ja erityisesti voikukalla, joiden terälehdet ovat todellisuudessakin pieniä ja monilukuisia, terälehtiä

kuvastavien avainpisteiden välisten kulmien keskiarvo S:mean_angle on pieni verrattuna lajeihin, joilla terälehtiä on vain muutama. Keskiarvo on suurin valkovuokolla, jonka isot ja harvalukuiset terälehdet ovat pitkulaisia. Leinikin ja orvokin terälehdet taas ovat muodoltaan pyöreämpiä. Näin ollen pienen lehtiluokan kulmamuuttujat näyttäisivät kuvastavan jossain määrin terälehtien muotoa lajeilla, joilla on harvassa isoja terälehtiä. Taulukosta 6 ja kuviosta 22b puolestaan havaitaan, että terälehtipisteiden välinen etäisyys S:mean_dist ei erottele yhtä selvästi lajeja toisistaan. Esimerkiksi voikukilla terälehtien kärkiä voi löytyä myös muualta kuin terälehtikehän reunalta (kuvio 17), mikä pidentää vierekkäisten terälehtipisteiden välistä etäisyyttä ja lisää siten muuttujan hajontaa.

Isot terälehdet erottuvat selkeimmin valkovuokolla. Niitä löytyi jossain määrin myös orvokeilta, mutta kukkien sisäinen värivaihtelu ja vuokkoja huonompi erottuminen taustasta harmaasävykuvassa heikensivät avainpisteiden löytymistä. Leinikeillä terälehdet ovat usein yhteenlimittyneitä, joten selkeitä erillisiä, terälehtien kokoisia pistemäisiä kohteita ei välttämättä ole löydettävissä. Voikukilla ja päivänkakkaroilla suuren terälehtiluokan avainpisteet kuvaavat esimerkiksi yhteenliittyneitä terälehtiryhmittymiä tai valoisuuseroja. Siten joidenkin lajien osalta on hankala tulkita, mitä suuren terälehtiluokan avainpisteet kuvastavat. Niiden tuottama informaatio ei välttämättä kuvasta lajien välisten piirteiden eroja, eikä niitä myöskään löydy kaikista kuvista. Taulukosta 6 havaitaan, että niillä ei ole kovin suurta merkitystä lajiluokan tunnistuksessa. Täten suuren terälehtiluokan avainpisteiden piirrevektorit voisi myös jättää analyyseista pois.



(a) Pisteiden välisten kulmien keskiarvo.

(b) Pisteiden välisten etäisyyksien keskiarvo.

Kuvio 22: Pienen terälehtiluokan avainpistemuuttujien keskiarvoja keskivirheineen.

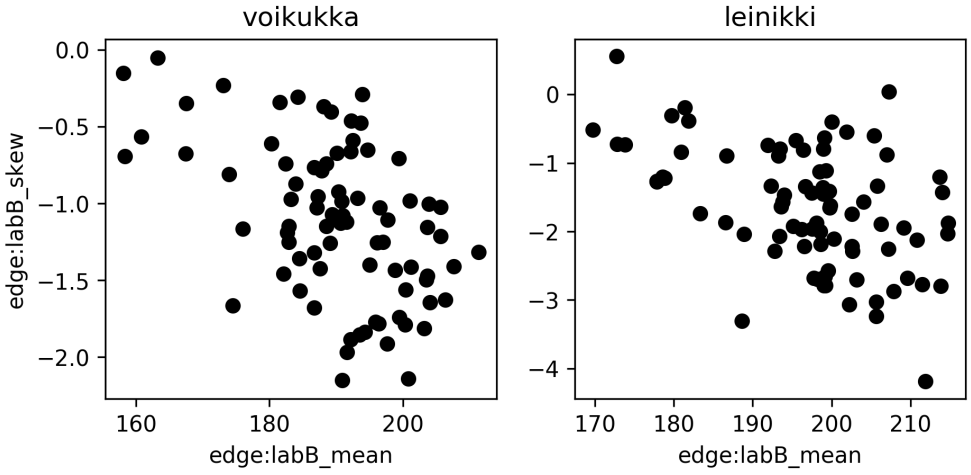
Taulukko 6: Piirremuuttujien vaikutus lajiluokkaan. Mittari $I_P(A_i;C)$ (kaava 3.7) kuvaa lajiluokan ja piirrevektorin yhteistä informaatiota ja mittari d_{χ^2} (kaava 3.6), lajiluokan ja piirrevektorin yhteisjakauman poikkeavuutta tilanteesta, jossa muuttujat ovat riippumattomia. Taulukko on lajiteltu laskevaan järjestykseen mittarin $I_P(A_i;C)$ mukaan.

Piirrevektori	$I_P(A_i;C)$	d_{χ^2}
edge:labB_mean	0,512	652,2
edge:labB_skew	0,467	587,6
edge:labB_sd	0,411	519,9
edge:labA_sd	0,395	510,4
S:sd_angle	0,356	453,2
cent:labB_mean	0,336	447,4
cent:labB_sd	0,320	408,8
S:mean_angle	0,318	416,6
cent:dist	0,293	332,1
edge:labA_mean	0,289	393,2
edge:dist	0,270	337,9
d1	0,258	334,4
cent:labB_skew	0,244	291,7
S2:hu1	0,230	299,9
d2	0,218	280,8
L:hu1	0,211	279,1
edge:hu4	0,201	237,4
cent:labA_skew	0,194	248,5

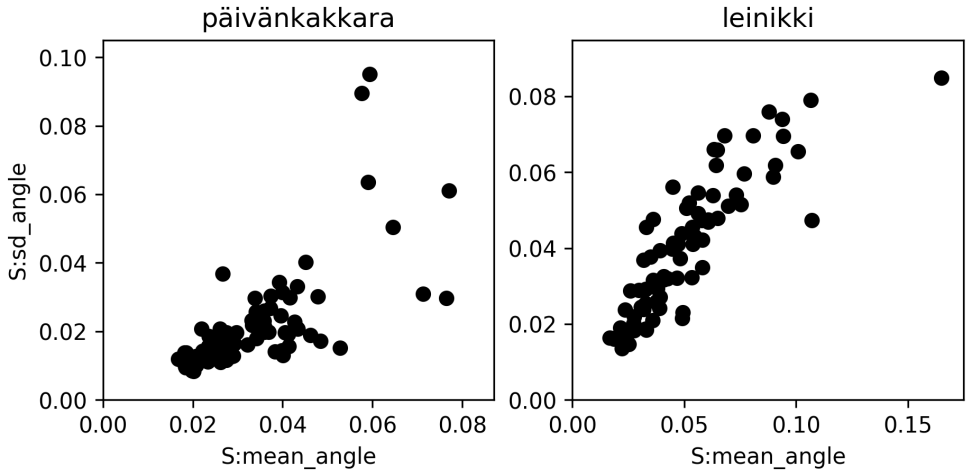
Piirrevektori	$I_P(A_i;C)$	d_{χ^2}
cent:labA_sd	0,190	244,2
d3	0,190	238,4
S:mean_dist	0,183	225,8
S:sd_dist	0,181	263,0
cent:hu4	0,176	218,6
cent:labA_mean	0,171	225,2
d5	0,152	162,7
L:mean_angle	0,140	164,3
edge:labA_skew	0,129	155,2
cent:hu2	0,127	147,8
d4	0,126	169,7
S1:hu1	0,126	137,2
edge:hu2	0,124	158,2
L:mean_dist	0,120	158,1
L:sd_angle	0,115	149,5
d6	0,105	132,1
L:sd_dist	0,102	122,2
d7	0,101	122,2

Mallin piirrevektorien määrä, 36, on jo melko suuri. Muuttujien määrän lisääntyessä kasvaa mahdollisuus, että malliin valitut muuttujat kuvaavat samaa asiaa tai muulla tavoin riippuvat toisistaan. Naiivi Bayes-menetelmä sisältää kuitenkin oletuksen havaittujen muuttujien riippumattomuudesta ehdolla luokkamuuttuja. Piirteiden korrelaatioita lajeittain tarkasteltaessa havaittiin itseisarvoltaan hyvin voimakkaitakin korrelaatioita. Esimerkiksi keltaisilla kukilla reunan väriä kuvaavat Lab:n b-kanavan keskiarvo edge:labB_mean ja vinousarvo

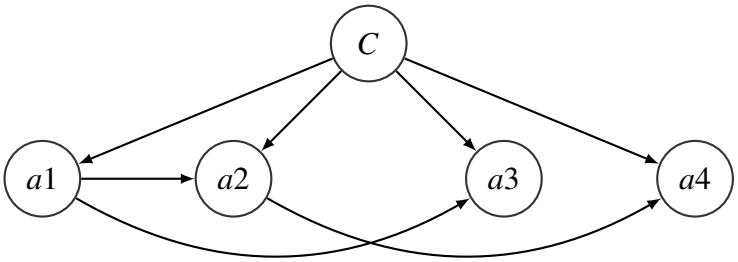
edge:labB_skew korreloivat negatiivisesti (kuvio 23). Pienen terälehtiluokan peräkkäisten pisteiden välisen kulman keskiarvo S:mean_angle ja keskihajonta S:sd_angle sen sijaan korreloivat positiivisesti (kuvio 24). Molemmat mainitut, korreloivat piirrevektorit vaikuttavat myös taulukon 6 perusteella paljon lajiluokan ennustamiseen. Kaikkia malliin valittuja piirrevektoreita ei siten voida pitää toisistaan riippumattomina, joten naiivista Bayes-mallista tulisikin siirtyä malliin, joka pystyisi ottamaan huomioon muuttujien välisiä riippuvuuksia.



Kuvio 23: Reunaosan Lab-väriavaruuden b-kanavan keskiarvo ja vinousarvo voikukalla ja leinikillä.



Kuvio 24: Pienen terälehtiluokan avainpisteiden välisten kulmien keskiarvot ja keskihajonnat päivänkakkaralla ja leinikillä.



Kuvio 25: Esimerkki puurakenteella täydennetystä naiivista Bayes-verkosta, jossa C on luokkasolmu.

4.5 Puu- tai metsärakenteella täydennetyt naiivit Bayes-mallit

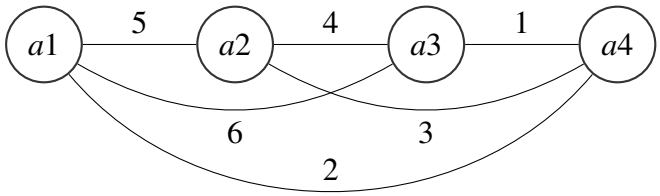
4.5.1 Mallien kuvaus

Piirremuuttujien välisiä riippuvuuksia voidaan mallintaa lisäämällä naiiviin Bayes-verkkoon kaaria piirremuuttujien välille. Tällaisia verkkoja kutsutaan täydennetyiksi naiiveiksi Bayes-verkoiksi (engl. *augmented naive Bayesian networks*; Friedman, Geiger ja Goldszmidt 1997). Parhaiten aineistoon sopiva täydellinen verkko ei kuitenkaan ole löydettävissä polynomisessa ajassa, minkä vuoksi täydennettyjen naiivien verkkojen rakennetta yleensä rajoitetaan (Friedman, Geiger ja Goldszmidt 1997). Puurakenteella täydennetty naiivi Bayes-malli (TAN-malli) on eräs tällainen verkkomalli. Mallissa jokaisella piirremuuttujalla on yhteys luokkasolmuun, ja piirremuuttujat yhtä lukuunottamatta riippuvat jostakin toisesta piirteestä — piirremuuttujilla on siten yhtä piirrettä lukuunottamatta kaksi vanhempaa: luokkasolmu ja jokin toinen piirremuuttuja (Friedman, Geiger ja Goldszmidt 1997). Mallin rakenne on esitetty kuviossa 25. Tavallisen naiivin Bayes-mallin tavoin luokkasolmu on koko verkon juurisolmu. Piirremuuttujat keskenään ilman luokkasolmua taas muodostavat puumaisen verkkorakenteen, mistä juontaa mallin nimi. Puurakenteen ansiosta mallin rakenne voidaan estimoida polynomisessa ajassa (Chow ja Liu 1968; Friedman, Geiger ja Goldszmidt 1997).

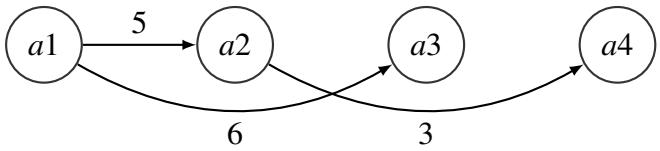
Chow ja Liu (1968) esittävät artikkelissaan algoritmin puumaisen Bayes-verkon muodostamiseksi. Friedman, Geiger ja Goldszmidt (1997) soveltavat Chow'n ja Liun (1968) algoritmia tilanteeseen, jossa mukana on myös luokkasolmu, eli puurakenteella täydennettyyn naiiviin Bayes-malliin. He myös osoittavat artikkelissaan algoritmin tuottaman rakenteen olevan

suurimman uskottavuuden estimaatti kuvatunlaisen rakennetyypin Bayes-verkolle. Algoritmi on seuraava:

1. Lasketaan jokaiselle piirrevektoriparille A_i ja A_j, $i \neq j$, ehdollinen yhteinen informaatio $I_P(A_i;A_j|C)$ ehdolla luokkamuuttuja C (ks. kaava 3.7).
2. Rakennetaan suuntaamaton verkko, jossa kaikki piirrevektorit ovat yhteydessä toisiinsa ja piirrevektoreiden A_i ja A_j välistä kaarta painotetaan yhteisellä informaatiolla $I_P(A_i;A_j|C)$. Kuviossa 26a on kuvattu kuvion 25 esimerkkiverkko tämän vaiheen jälkeen.
3. Muodostetaan täysyhteydellisestä verkosta esimerkiksi Kruskalin algoritmia (Kruskal 1956) hyödyntäen suurin virittävä puu.
4. Muokataan suuntaamaton puumainen verkko suunnatuksi valitsemalla juurisolmuksi jokin piirremuuttujista ja suuntaamalla jatkossa kaikki kaaret valitusta solmusta poispäin. Kuviossa 26b on kuvattu kuvion 25 esimerkkiverkko tämän vaiheen jälkeen.
5. Otetaan luokkasolmu C mukaan malliin yhdistämällä se suunnatuilla kaarilla jokaiseen piirremuuttujaan A_i (kuvio 25).



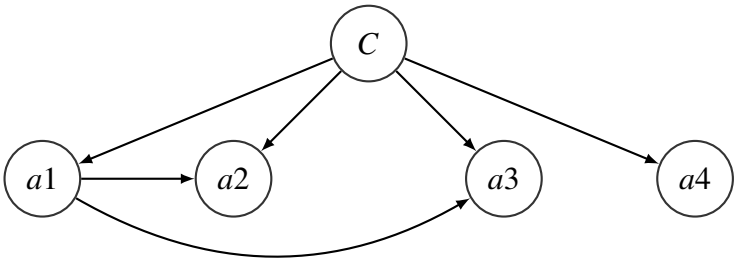
(a) Täysyhteydellinen suuntaamaton verkko



(b) Puuverkko, jonka juurisolmuna on $a1$

Kuvio 26: TAN-verkon rakennusalgoritmin vaiheita 2 ja 4 vastaavat verkot. Kaarien arvot vastaavat muuttujien yhteisen informaation määrää ehdolla luokkamuuttuja.

Puurakenteella täydennetty naiivi Bayes-malli voidaan edelleen muokata verkoksi, joka luokkasolmu poislukien muodostaakin puun sijaan metsän (Lucas 2002; Jayech ja Mahjoub



Kuvio 27: Esimerkki metsärakenteella täydennetystä naiivista Bayes-verkosta, jossa C on luokkasolmu.

2010). Tällöin jokaisella piirremuuttujalla on korkeintaan yksi vanhempi, mutta puurakenteisesta mallista poiketen vanhemman olemassaolo ei ole ehdoton (kuvio 27). Metsärakenteella täydennetty naiivi Bayes-malli (FAN-malli) on siten TAN-mallia joustavampi.

Jayech ja Mahjoub (2010) esittävät artikkelissaan erään algoritmin metsän sisältävän Bayes-verkon muodostamiseksi. Algoritmi seuraa pääpiirteissaan TAN-mallin rakentamisessa käytettävää algoritmia, mutta sisältää muutaman täydennyksen. Piirremuuttujien välisten ehdollisten yhteisten informaatioiden $I_P(A_i;A_j|C)$ lisäksi lasketaan myös luokkamuuttujan ja kunkin piirrevektorin välinen yhteisinformaatio $I_P(A_i;C)$. Puumaisen suunnatun verkon juurisolmuksi valitaan piirremuuttuja, jolla on suurin yhteinen informaatio luokkamuuttujan kanssa eli suurin vaikutus luokkamuuttujaan. Tämän jälkeen valitaan jokin raja-arvo ja poistetaan kaikki piirrevektoreiden väliset kaaret, joiden paino on raja-arvoa pienempi. Näin piirrevektorit muodostavat puun sijaan metsän. Jayech ja Mahjoub (2010) käyttivät artikkelissaan ensisijaisena raja-arvona piirremuuttujien yhteisten informaatioarvojen keskiarvoa.

4.5.2 Mallien rakentaminen

TAN- ja FAN-mallit rakennettiin luvussa 4.5.1 kuvatuilla algoritmeilla. Verkkorakenteen muodostamisessa tarvittava suurin virittävä puu laskettiin Pythonin scipy-kirjaston harvoina matriiseina esitettäville verkoille tarkoitetulla pienin virittävä puu -algoritmilla[14] vaihtamalla painot negatiivisiksi. Mallien parametrit estimoitiin kaavalla 3.3 olettaen tasaiset priorit. Puuttuvat havainnot käsiteltiin jättämällä jakaumaa muodostettaessa pois laskuista yksilöt,

14. Ks. scipy-dokumentaatio: minimum_spanning_tree

joilta havainto tai muuttujan vanhemman havainnot puuttuivat. Yhteistä informaatiota laskettaessa käytettiin niinikään kaikkia mahdollisia havaintoja.

Täydennettyjen naiivien Bayes-mallien rakenteet on esitetty taulukossa 7. Piirremuuttujat sijaitsevat taulukon riveillä. Sarakkeilla sijaitsevat eri mallit, joten kustakin sarakkeesta on luettavissa kyseisen mallin rakenne. Taulukon soluihin on merkitty kunkin muuttujan kohdalle sen piirremuuttujavanhempi.

TAN-verkon puuosion juurisolmuksi valittiin pienien terälehtien välisten kulmien keskihajonta S:sd_angle, koska se sisälsi eniten riippuvuuksia toisiin piirremuuttujiin — muuttujasta muodostui 16 muun piirremuuttujan vanhempisolmu. Tällaisella alkusolmulla saatiin verkko, joka sisältää taulukossa 7 symbolilla TAN kuvatut piirremuuttujien väliset yhteydet. Taulukosta 7 havaitaan, että keskihajonta on yhteydessä sekä väriä että muotoa kuvaaviin piirrevektoreihin. Pienien terälehtien välisten kulmien keskihajonnan ohella myös vastaavalla keskiarvo-piirrevektorilla S:mean_angle on melko paljon yhteyksiä toisiin, muotoa kuvaaviin piirrevektoreihin. Muita vanhempisolmuja ovat suurien terälehtien kulmamuuttujat L:mean_angle ja L:sd_angle, pienien terälehtien etäisyyksien keskiarvo S:mean_dist ja kukan reunaosan väriä kuvaavat muuttujat. Suurella osalla näistä on yhteyksiä sekä väriä että muotoa kuvaaviin piirrevektoreihin.

FAN-verkon juurisolmuksi valikoitui reunaosan b-värikanavan keskiarvo edge:labB_mean, koska lasketun yhteisen informaation mukaan kyseinen piirre vaikutti eniten luokkamuuttujaan (taulukko 6). Käytettäessä piirremuuttujien yhteisinformaatioiden keskiarvoa mukaan valittavien kaarien painojen raja-arvona kävi kuitenkin niin, että kaikkien kaarien painojen arvo oli raja-arvoa suurempi. Piirremuuttujat muodostivat siten edelleen keskenään puun metsän sijaan, eikä malli juurikaan poikennut TAN-mallista — erona pienien terälehtien välisten kulmien keskihajonta S:sd_angle on siirtynyt reunaosan b-värikanavan keskiarvon edge:labB_mean lapseksi, muutoin malli on sama. Taulukossa 7 tämän mallin symboli on FAN-avg. Keskiarvo ei välttämättä ole hyvä raja-arvo, kun piirremuuttujia on niinkin paljon kuin 36 kappaletta. Tällöin yhteyksiä piirremuuttujien välille voidaan muodostaa kaikkiaan $36 * 35/2 = 630$ kappaletta, mutta verkossa yhteyksien määrä on vain noin kahdeskymmenesosa tästä, 35 kappaletta. Esimerkkikuvion 26 neljän solmun verkossa keskiarvo toimisi paremmin: solmujen $a2$ ja $a4$ välinen kaari poistettaisiin. Täten keskiarvon lisäksi raja-

Taulukko 7: Yhteydet piirremuuttujien välillä TAN- ja FAN-malleissa. Riveillä sijaitsevat piirremuuttujat. Sarakkeilla sijaitsevat eri mallit, joten kustakin sarakkeesta on luettavissa kyseisen mallin rakenne ja sen muuttujat. Mallien nimilyhenteet on avattu tekstissä. Taulukon soluihin on merkitty kunkin muuttujan kohdalle sen vanhempisolmu. Jos solu on tyhjä, kyseisen rivin piirremuuttujan ainoa vanhempi on lajiluokka ko. mallissa.

	Malli				
Muuttuja	TAN	FAN-avg	FAN-90	FAN-95	FAN-99
S:mean_dist	S:sd_angle	S:sd_angle			
S:sd_dist	S:mean_dist	S:mean_dist			
S:mean_angle	S:sd_angle	S:sd_angle	S:sd_angle	S:sd_angle	S:sd_angle
S:sd_angle		edge:labB_mean	edge:labB_mean	edge:labB_mean	edge:labB_mean
L:mean_dist	L:mean_angle	L:mean_angle			
L:sd_dist	L:sd_angle	L:sd_angle			
L:mean_angle	edge:labA_mean	edge:labA_mean	edge:labA_mean		
L:sd_angle	edge:labB_mean	edge:labB_mean	edge:labB_mean		
S1:hu1	S:sd_angle	S:sd_angle			
S2:hu1	S:mean_angle	S:mean_angle	S:mean_angle	S:mean_angle	
L:hu1	S:sd_angle	S:sd_angle			
d1	S:sd_angle	S:sd_angle	S:sd_angle	S:sd_angle	
d2	S:mean_angle	S:mean_angle	S:mean_angle	S:mean_angle	
d3	S:mean_angle	S:mean_angle	S:mean_angle	S:mean_angle	
d4	S:sd_angle	S:sd_angle	S:sd_angle		
d5	S:sd_angle	S:sd_angle	S:sd_angle	S:sd_angle	
d6	S:sd_angle	S:sd_angle	S:sd_angle	S:sd_angle	
d7	S:mean_angle	S:mean_angle	S:mean_angle		
cent:dist	S:mean_angle	S:mean_angle	S:mean_angle		
edge:dist	S:sd_angle	S:sd_angle	S:sd_angle	S:sd_angle	
cent:labA_mean	L:sd_angle	L:sd_angle	L:sd_angle		
cent:labA_sd	edge:labB_mean	edge:labB_mean	edge:labB_mean		
cent:labA_skew	S:sd_angle	S:sd_angle	S:sd_angle	S:sd_angle	
cent:labB_mean	S:sd_angle	S:sd_angle	S:sd_angle	S:sd_angle	S:sd_angle
cent:labB_sd	S:sd_angle	S:sd_angle	S:sd_angle	S:sd_angle	
cent:labB_skew	S:sd_angle	S:sd_angle	S:sd_angle	S:sd_angle	
cent:hu2	L:sd_angle	L:sd_angle			
cent:hu4	S:mean_angle	S:mean_angle	S:mean_angle		
edge:labA_mean	L:sd_angle	L:sd_angle	L:sd_angle	L:sd_angle	
edge:labA_sd	edge:labB_mean	edge:labB_mean	edge:labB_mean	edge:labB_mean	edge:labB_mean
edge:labA_skew	S:sd_angle	S:sd_angle	S:sd_angle	S:sd_angle	S:sd_angle
edge:labB_mean	S:sd_angle				
edge:labB_sd	edge:labA_sd	edge:labA_sd	edge:labA_sd	edge:labA_sd	edge:labA_sd
edge:labB_skew	S:sd_angle	S:sd_angle	S:sd_angle	S:sd_angle	
edge:hu2	edge:labA_sd	edge:labA_sd			
edge:hu4	edge:labA_sd	edge:labA_sd	edge:labA_sd		

Taulukko 8: Puurakenteella täydennetyn naiivin Bayes-mallin sekaannusmatriisi (%).

	Päivänkakkara	Voikukka	Leinikki	Valkovuokko	Orvokki
Päivänkakkara	85,0	1,3	2,5	8,8	2,5
Voikukka	3,8	90,0	6,3	0,0	0,0
Leinikki	5,0	1,3	77,5	8,8	7,5
Valkovuokko	7,5	0,0	2,5	90,0	0,0
Orvokki	6,3	0,0	17,5	13,8	62,5

arvoksi kokeiltiin piirremuuttujien yhteisinformaation persentiilejä 90, 95 ja 99. Nämä on nimetty FAN-90-, FAN-95- ja FAN-99-malleiksi, ja niiden sisältämät piirremuuttujien väliset yhteydet on kuvattu taulukossa 7 vastaavilla mallisymboleilla. FAN-90-mallissa jäljelle jää 27 yhteyttä ja FAN-95-mallissa 18 yhteyttä; FAN-99-mallissa yhteyksiä on enää kuusi. FAN-95- ja FAN-99-malleissa yhteyksiä jää pääasiassa sellaisten piirteiden välille, joilla on taulukon 6 perusteella myös paljon vaikutusta luokkamuuttujaan.

4.5.3 Luokittelutulokset

Puu- tai metsärakenteella täydennettyjen naiivien Bayes-mallien sopivuutta aineistoon tarkasteltiin jälleen *leave one out* -ristiinvalidoinnilla. Taulukossa 8 on esitetty TAN-mallin sekaannusmatriisi. Vertailtaessa naiivin Bayes-mallin tuloksia (taulukko 5) TAN-mallin tuloksiin (taulukko 8) havaitaan, että luokittelutulokset ovat huonontuneet kaikissa lajiluokissa. Eniten luokittelutarkkuus on huonontunut orvokilla, jopa 35 prosenttiyksikköä. Myös leinikki tunnistetaan aiempaa selvästi huonommin. Muilla lajeilla muutokset luokittelutarkkuudessa ovat joitakin prosenttiyksikköjä. Mielenkiintoista kuitenkin on, että TAN-mallissa leinikit luokittuvat naiivia Bayes-mallia harvemmin voikukkien luokkaan. Voikukat sen sijaan luokittuvat naiivia Bayes-mallia useammin leinikkien luokkaan.

Jotta vertailu eri FAN-mallien välillä olisi helpompaa, kaikkien mallien luokittelutarkkuudet on koottu taulukkoon 9. Mallikohtaiset sekaannusmatriisit on esitetty liitteessä B. Vertailtaessa FAN-avg-mallin tuloksia TAN-mallin tuloksiin havaitaan, että luokittelutulokset ovat keskimääräisesti parantuneet, kun juurisolmuksi on vaihdettu satunnaisen solmun sijaan sol-

mu, jolla on eniten vaikutusta luokkamuuttujaan. TAN-mallista FAN-malliin siirtyminen vaikutti eniten leinikkiin ja orvokkiin. Päivänkakkaroiden ja valkovuokkojen luokittelutulokset huononivat hieman, mutta ero on vain kuvan tai kahden luokkaa. Vertailtaessa keskenään erilaisia FAN-malleja havaitaan puolestaan, että luokittelutulokset pääsääntöisesti paranivat, kun yhteyksiä piirremuuttujien välillä vähennettiin. Orvokkia lukuunottamatta lajikohtaiset erot FAN-90-, FAN-95- ja FAN-99-mallien välillä ovat kuitenkin erittäin pieniä.

Taulukko 9: TAN- ja FAN-mallien luokittelutarkkuudet (%) lajeittain ja kaikkien lajien keskiarvo.

	Päivänkakkara	Voikukka	Leinikki	Valkovuokko	Orvokki	Keskiarvo
TAN	85,0	90,0	77,5	90,0	62,5	81,0
FAN-avg	83,8	91,3	86,3	86,3	78,8	85,3
FAN-90	86,3	95,0	87,5	92,5	81,3	88,5
FAN-95	86,3	95,0	86,3	90,0	92,5	90,0
FAN-99	87,5	97,5	88,8	92,5	95,0	92,3

4.5.4 Havaintoja menetelmän ominaisuuksista ja käyttökelpoisuudesta

Jo muotopiirteillä vahvistetun naiivin Bayes-mallin tuloksia tarkasteltaessa havaittiin, että merkittävästi lajiluokkaan ja siten tunnistukseen vaikuttavien piirremuuttujien välillä on havaittavissa riippuvuutta (taulukko 6; kuviot 23 ja 24). FAN-95- ja FAN-99-mallien rakenteet vahvistavat tätä havaintoa, sillä malleissa yhteyksiä jää pääasiassa sellaisten piirteiden välille, joilla on taulukon 6 perusteella myös paljon vaikutusta luokkamuuttujaan. Piirremuuttujat voisi siten karkeasti jaoteltuna jakaa kahteen luokkaan: ensimmäiseen luokkaan kuuluisivat muuttujat, jotka riippuvat paitsi lajiluokasta myös toisista piirremuuttujista, ja toiseen luokkaan kuuluisivat muuttujat, jotka ovat riippumattomia toisista piirremuuttujista ja joiden vaikutus lajiluokkaan on vähäinen. Voitaisiinkin pohtia, tulisiko toiseen luokkaan kuuluvia piirremuuttujia poistaa mallista, ja parantaisiko toimenpide luokittelutuloksia.

TAN- ja FAN-malleja voidaan rakentaa eri tavoin. Keogh ja Pazzani (1999) lisäsivät malliin kaaria luokittelutarkkuuteen pohjautuen: piirremuuttujien välille lisättiin kaari, jos luokit-

telutarkkuus parani, muutoin palautettiin edeltävä malli. Keoghin ja Pazzanin (1999) luokittelutarkkuuteen pohjautuva algoritmi antoi Friedmanin, Geigerin ja Goldszmidtin (1997) TAN-mallia parempia luokittelutuloksia. Muuttujien yhteisinformaatioon pohjautuva algoritmi oli kuitenkin laskennallisesti helpompi toteuttaa, ja lopulta erot eri mallien luokittelutarkkuuksissa olivat keskimäärin vain kolmen prosenttiyksikön luokkaa (Keogh ja Pazzani 1999). Hamine ja Helman (2005) sekä Lucas (2002) hyödynsivät FAN-malleissaan piirremuuttujien yhteisinformaation pohjalle rakennettua algoritmia, mutta poistivat ensin raja-arvon alapuolelle jäävät kaaret ja vasta sen jälkeen muodostivat kullekin osaverkolle suurimman virittävän puun ja valitsivat osaverkoille juurisolmut. Itse asiassa Lucas (2002) ei määrännyt kaarien määrää yhteisinformaatioon perustuvan raja-arvon pohjalta, vaan etsi kullekin aineistolle sopivan kaarien määrän luokittelutarkkuuteen perustuen. Täten Lucasin (2002) algoritmi oli eräänlainen sekoitus yhteisinformaatioon ja luokittelutarkkuuteen pohjautuvia algoritmeja.

Tässä tutkimuksessa saadut tulokset ovat hyvin pitkälti yhteneväisiä Jayechin ja Mahjoubin (2010) saamien tulosten kanssa. Jayechin ja Mahjoubin (2010) havaintoaineistossa naiivi Bayes-malli antoi parempia luokittelutuloksia kuin TAN-malli tai FAN-avg-malli. Ainoastaan FAN-malli, jonka kaarien valinnassa käytettyä raja-arvoa oli tiukennettu piirteiden yhteisinformaation keskiarvosta, antoi naiivia Bayes-mallia parempia luokittelutuloksia. Artikkelissa ei kuitenkaan täsmällisesti mainittu, miten tämä raja-arvo oli saatu. Lucas (2002) tutki artikkelissaan järjestelmällisemmin kaarien määrän lisäämisen vaikutusta luokittelutarkkuuteen sekä mallin ennustaman luokan posterioritodennäköisyyteen. Posterioritodennäköisyyden entropian avulla pystytään kuvaamaan, kuinka suuri mallin ennustaman luokan todennäköisyys keskimäärin on verrattuna muiden luokkien todennäköisyyteen. Lucasin (2002) tutkimuksen mukaan riippuu paljon aineistosta, vaikuttaako kaarien määrän lisäys luokittelutarkkuuteen. Posterioritodennäköisyyteen kaarien määrällä oli jonkin verran enemmän vaikutusta. Luokkien posterioritodennäköisyyksien laskeminen luokittelutarkkuuden ohella voisi siten tuoda lisää informaatiota mallin sopivuudesta aineistoon.

Jayechin ja Mahjoubin (2010) mallissa piirrevektoreita oli yhdeksän ja kunkin piirteen arvot oli jaettu tutkimuksesta riippuen joko viiteen tai kymmeneen luokkaan. Siten sekä piirteiden että piirteiden sisäisten luokkien määrä oli huomattavasti pienempi kuin tässä tutkimukses-

sa. Jayechin ja Mahjoubin (2010) tutkimuksessa piirrevektoreiden arvojen jako kymmeneen luokkaan antoi parempia luokittelutuloksia kuin jako viiteen luokkaan. He myös kokeilivat naiiviin Bayes-malliin erisuuruisia luokkajakoja ja totesivat jaon kahdeksaan luokkaan antavan parhaan tuloksen.

Tässä tutkimuksessa kunkin piirremuuttujan arvot on jaettu 20 luokkaan. Määrä on suuri ottaen huomioon, että havaintoja kussakin lajiluokassa on vain 80. Kun piirteellä on lajiluokan lisäksi piirremuuttuja toisena vanhempisolmuna, vain enää keskimäärin joka viidennessä piirremuuttujan luokassa on havainto. Datan määrä menee niin pieneksi, ettei kunnon jakaumatietoa ole enää saatavilla eikä piirteen todennäköisyysjakauman parametriestimaatteja voida siten pitää enää luotettavina (Friedman, Geiger ja Goldszmidt 1997; Lucas 2002). Sekä Friedman, Geiger ja Goldszmidt (1997) että Lucas (2002) tasoittivat artikkelissaan todennäköisyysparametria $P(X_i|\Pi_{X_i})$ prioritiedolla painottamalla sitä enemmän prioritietoa, mitä vähemmän dataa oli saatavilla. Friedman, Geiger ja Goldszmidt (1997) käyttivät priorina piirremuuttujan X_i marginaalijakaumaa. Tasoitetun parametrin TAN-malli antoi tasoittamattoman parametrin mallia paremman luokittelutuloksen erityisesti aineistoissa, joissa havaintoja oli vähän mutta tunnistettavia luokkia paljon (Friedman, Geiger ja Goldszmidt 1997). Suuressa osassa testiaineistoja ero oli kuitenkin hyvin pieni. Tasoitetun parametrin käyttö ei kuitenkaan todennäköisesti riittäisi tässä pro gradu -työn aineistossa, sen verran vähän dataa jää piirremuuttujan luokkiin. Käytännössä parametrien määrä on niin suuri suhteessa havaintojen määrään, että on hankala sanoa, mitä parametriestimaatit edustavat. TAN-mallissa suurin osa parametriestimaateista vastaa tilannetta, jossa luokkakombinaatiossa ei ole tehty yhtään havaintoa.

Erityisesti TAN-mallin tuottama verkko lieneekin aivan liian monimutkainen näin pieneen aineistoon. Tulkintaa tukee se, että vähennettäessä piirremuuttujien välisiä riippuvuuksia luokittelutulokset paranevat. Parhaimman luokittelutuloksen antava FAN-99-malli ei kuitenkaan sekään yllä vastaavan naiivin Bayes-mallin tasolle. Yhtälailla kaikissa TAN- ja FAN-malleissa on kahden vanhemman ja vähäisen datan määrän aiheuttama epäluotettavien estimaattien ongelma. Muuttujien välisiä riippuvuuksia tulisikin hallita tässä aineistossa jollakin muulla menetelmällä. Myös piirremuuttujien luokkien määrää voisi pienentää. Avoimeksi kysymykseksi kuitenkin jää, mikä olisi sopiva luokkien määrä.

4.6 Rakenteeltaan rajoittamattomien verkkojen mallit

4.6.1 Mallin kuvaus ja rakentaminen

Rakenteeltaan rajoittamattomissa Bayes-verkoissa paras rakenne valitaan sopivuuskriteerin perusteella (Friedman, Geiger ja Goldszmidt 1997). Malleissa luokkamuuttuja ei välttämättä ole verkon juurisolmu eikä muuttujien välisiä yhteyksiä ole muutoinkaan ennalta rajoitettu (Friedman, Geiger ja Goldszmidt 1997). Kyseessä on siten Bayes-verkon yleinen muoto. Estimoinnin yksinkertaistamiseksi voi kuitenkin olla syytä rajoittaa piirresolmun vanhempien määrää, joskin yleensä mallin ja aineiston yhteensopivuutta mittaavat kriteerit huolehtivat siitä, ettei mallista tule liian monimutkaista (Myllymäki ja Tirri 1998). Rajoittamattomiin malleihin voi tulla mukaan myös hierarkkisia rakenteita, mikäli kaikki piirteet eivät ole suoraan yhteydessä luokkasolmuun.

Ensimmäiset rakenteeltaan rajoittamattomat mallit kokeiltiin muodostaa muotopiirteillä vahvistetun mallin pohjalta. Mallien rakenteet estimoitiin Pythonin pgmpy-kirjaston *hill climbing* -algoritmilla[15] (ks. luku 3.3.2). Verkkojen aloitusrakenteiksi kokeiltiin sekä muotopiirteillä vahvistetun mallin mukaista naiivia Bayes-verkkoa että tyhjää verkkoa. Ensimmäiset rajoittamattomat mallit sisälsivät kaikki 36 piirrevektoria ja lajiluokan. Rajoittamattomia malleja kokeiltiin muodostaa myös siten, että malleista jätettiin pois ison terälehtiluokan avainpisteiden piirteet. Ison terälehtiluokan piirremuuttujilla ei taulukon 6 mukaan ole juurikaan vaikutusta lajiluokkaan, ja lisäksi niistä puuttui havaintoja. Mallin ja datan yhteensopivuutta mitattiin joko K2- tai BIC-sopivuuskriteerillä[16] (ks. luku 3.3.2; Cooper ja Herskovits 1992; Heckerman 1999). Mallin rakenteeksi valittiin verkko, jonka sopivuuskriteeri oli suurin. Samalla pystyttiin vertaamaan, tuottivatko eri kriteerit erilaisen verkon. Puuttuvat havainnot käsiteltiin rakenne-estimoinnissa muuttujittain, toisin sanoen jokaisen muuttujan osalta hyödynnettiin kaikkia mahdollisia havaintoja eli niitä, joissa muuttujasta itsestään tai sen vanhemmista ei puuttunut arvoja. Jotta rakenne-estimointi pysyisi yksinkertaisena eikä havaintojen määrä piirremuuttujan luokkaa kohti kävisi liian pieneksi, kullakin solmulla sai olla korkeintaan yksi vanhempi.

15. Ks. pgmpy-dokumentaatio: HillClimbSearch
16. Ks. pgmpy-dokumentaatio: estimators

TAN- ja FAN-mallien yhteydessä (ks. luku 4.5.4) pohdittiin piirrevektoreiden luokkien isohkoa määrää ja sen vaikutusta parametriestimaattien luotettavuuteen. Tämän vuoksi rajoittamattomia malleja kokeiltiin rakentaa myös piirrevektoreista, joiden luokkien määrä oli 20:n sijaan 5. Lukumäärä 5 on lähinnä valistunut arvaus. Se on riittävän suuri muodostamaan histogrammin mutta samalla riittävän pieni, jotta piirremuuttujalle voidaan sallia kaksi vanhempisolmua. Kahden 5-luokkaisen vanhemman tapauksessa 5-luokkaisen piirremuuttujan luokkaa kohti jää keskimäärin noin 3 havaintoa, $400/(5 \times 5)/5 = 3,2$. Viisiluokkaisten piirremuuttujien malleissa ei ollut mukana ison terälehtiluokan piirrevektoreita. Mallit olivat siten selvästi yksinkertaisempi versio alkuperäisestä 36 piirremuuttujan mallista.

Kun mallin rakenne oli saatu estimoitua, parametrit estimoitiin kaavassa 3.5 kuvatulla tavalla. Piirremuuttujien jakaumia muodostettaessa puuttuvat havainnot käsiteltiin vastaavasti kuin rakennetta estimoitaessa: jakaumaa muodostettaessa jätettiin pois laskuista yksilöt, joilta havainto tai muuttujan vanhemman havainnot puuttuivat.

4.6.2 Mallien rakenteet ja luokittelutulokset

Taulukossa 10 on esitetty rajoittamattomien mallien rakenteet. Kustakin sarakkeesta on luettavissa kyseisen mallin rakenne ja sen muuttujat. Solujen harmaasävyt kuvaavat rivin piirremuuttujan asemaa kyseisessä mallissa. Valkoinen väri solussa tarkoittaa, että piirremuuttuja kuuluu luokkamuuttujan Markovin peitteeseen (engl. *Markov blanket*). Toisin sanoen piirremuuttuja on joko luokkamuuttujan vanhempi, lapsi tai luokkamuuttujan lapsen vanhempi; vain nämä piirteet vaikuttavat luokkamuuttujaan (Friedman, Geiger ja Goldszmidt 1997). Jos solu on tummanharmaa, kyseisessä mallissa ei ole ollut mukana rivin piirremuuttujaa. Taulukon symbolit on kuvattu tarkemmin taulukon otsikkotekstissä.

Mallien nimilyhenteet ovat taulukossa 10 seuraavat: Etuliite *K2* / *BIC* kertoo mallin sopivuuskriteerin (ks. luku 3.3.2). Loppuliite *naive* / *empty* kertoo, onko rakenteen estimoinnissa lähdetty liikkeelle tyhjästä mallista (*empty*) vai naiivista mallista. Jos kyseinen tieto puuttuu nimestä, alkumallina on ollut samaan kategoriaan luettavissa malleissa aina tyhjä malli. Pieni *b* ilmaisee, että mallista on poistettu ison terälehtiluokan piirrevektorit. Loppuliite *C5* kertoo, että mallin piirremuuttujat ovat 5-luokkaisia alkuperäisten 20-luokkaisten sijaan.

C5-malleissa numero *1* / *2* ilmaisee, kuinka monta vanhempaa piirremuuttujalle on sallittu rakenne-estimoinnissa.

Naiivin mallin pohjalta rakennetun K2-sopivuuskriteerillä estimoidun mallin K2-naive rakenne ei kovin paljon poikkea naiivista Bayes-verkosta. Ainoastaan suurta terälehtiluokkaa edustavien muuttujien alle muodostuu lapsisolmuja, ja toisaalta lajiluokka siirtyy suurien terälehtien välisen kulman keskiarvon L:mean_angle alle. Jos K2-estimoinnissa lähdetään liikkeelle tyhjästä mallista K2-empty, suurta terälehtiluokkaa edustavien muuttujien alle muodostuu entistä enemmän lapsisolmuja. Lisäksi yhteyksiä muodostuu esimerkiksi pienen terälehtiluokan muuttujien välille. Taulukkoon 14 on koottu suurimpia piirremuuttujien välisiä korrelaatioita. Taulukosta voidaan havaita, että terälehtimuuttujien välillä on todellinen yhteys. Niiden yhteys on myös sisällöllisesti tulkittavissa. Sen sijaan deskriptorien, Hun momenttien ja väriä kuvaavien muuttujien yhteydet suuren terälehtiluokan muuttujiin ovat sisällöllisesti hankalammin tulkittavissa. Näiden piirrevektoreiden siirtyminen lajisolmun alta suuren terälehtiluokan solmujen lapsiksi on todennäköisesti selitettävissä sillä, että kyseisillä piirrevektoreilla on keskimäärin vähemmän merkitystä lajintunnistuksessa kuin luokkasolmun alle jääneillä (taulukko 6). Vaikutusta saattaa olla myös sillä, että suuren terälehtiluokan piirremuuttujista puuttuu havaintoja. Huomattakoon kuitenkin, että vaikka reunaosan Lab:n b-kanavan keskiarvo edge:labB_mean on suuren terälehtiluokan piirteen L:sd_dist alla, se on samalla myös lajiluokan vanhempisolmu ja siten suoraan yhteydessä lajiluokkaan.

Jos K2-kriteerillä estimoidusta mallista jätetään pois suuren terälehtiluokan piirrevektorit (malli K2b-empty), saadaan edellisiä malleja sisällöllisesti järkevämmän oloisia yhteyksiä piirremuuttujien välille — yhteyksiä muodostuu esimerkiksi momenttimuuttujien välille sekä pienen terälehtiluokan piirteiden välille (taulukko 10). K2b-empty-mallista jäävät suuren terälehtiluokan piirteiden ohella pois myös deskriptorimuuttujat d6 ja d7 sekä keskiosan pienen skaalan avainpisteiden momenttimuuttuja S1:hu1. Taulukon 6 perusteella nämä eivät juurikaan vaikuta lajintunnistukseen. Jos kaikki piirremuuttujat sisältävän mallin rakenne estimoidaan BIC-kriteerillä, malliin tulee mukaan vain muutama väriä kuvaava piirremuuttuja. Tosin taulukon 6 perusteella piirteet ovat juuri niitä, jotka eniten vaikuttavat lajintunnistukseen.

Taulukko 10: Rakenteeltaan rajoittamattomien mallien yhteydet lajiluokkaan ja toisiin piirremuuttujiin. Riveillä sijaitsevat muuttujat. Sarakkeilla sijaitsevat eri mallit, joten kustakin sarakkeesta on luettavissa kyseisen mallin rakenne ja sen muuttujat. Mallien nimilyhenteet on avattu tekstissä. Soluihin on merkitty kunkin muuttujan kohdalle sen vanhempisolmu. Merkki *X* tarkoittaa, että piirteen vanhempi on luokkasolmu. Merkki *O* tarkoittaa, että kyseinen muuttuja on koko verkon juurisolmu. Valkoinen väri solussa tarkoittaa, että ko. rivin piirteellä on riippuvuus luokkamuuttujaan. Jos solu on tummanharmaa, kyseisessä mallissa ei ole mukana rivin piirremuuttujaa. Vaaleanharmaa väri solussa tarkoittaa, että rivin piirremuuttuja on mukana mallissa, mutta sillä ei ole suoraa riippuvuutta luokkamuuttujan kanssa.

	Malli						
Muuttuja	K2-naive	BIC-naive	K2-empty	K2b-empty	K2b-2C5	K2b-1C5	BICb-2C5
lajiluokka	L:mean_angle	**O**	edge:labB_mean	edge:labB_mean	edge:labB_mean cent:labB_skew	edge:labB_mean	cent:labB_mean
S:mean_dist	X		L:mean_dist	S:sd_dist	S:sd_dist S:mean_angle	S:sd_dist	S:sd_dist
S:sd_dist	X		S:mean_dist	X	X	X	X
S:mean_angle	X		S:sd_angle	S:sd_angle	S:sd_angle	S:sd_angle	X
S:sd_angle	X		X	X	X	X	S:mean_angle
L:mean_dist	X		L:mean_angle				
L:sd_dist	L:sd_angle		**O**				
L:mean_angle	**O**		L:sd_angle				
L:sd_angle	L:mean_angle		L:sd_dist				
S1:hu1	L:mean_angle		L:mean_angle		X	X	
S2:hu1	X		X	X	S:sd_angle	S:sd_angle	S:sd_angle
L:hu1	X		L:mean_angle	X	X	X	X
d1	X		X	X	X	X	X
d2	X		X	X	X	X	X
d3	X		X	X	X	X	X
d4	X		L:mean_dist	X	X	X	X
d5	X		L:sd_angle	X	X	X	X
d6	L:mean_angle		L:mean_angle		d3 S:mean_angle	S:mean_angle	
d7	L:mean_dist		L:mean_dist		X	X	
cent:dist	X		edge:dist	edge:dist	X	X	X
edge:dist	X		X	X	cent:dist cent:labB_sd	cent:dist	cent:dist
cent:labA_mean	X		L:mean_angle	X	X edge:labA_mean	edge:labA_mean	edge:labA_mean
cent:labA_sd	X		L:sd_dist	X	X cent:labA_mean	X	X
cent:labA_skew	X		L:sd_dist	X	X	X	X
cent:labB_mean	X		X	X	cent:labB_skew	cent:labB_skew	cent:labB_skew
cent:labB_sd	X		X	X	X	X	X
cent:labB_skew	X		X	X	**O**	**O**	**O**
cent:hu2	X		L:sd_angle	cent:hu4	X	X	
cent:hu4	X		L:mean_dist	X	cent:hu2	cent:hu2	
edge:labA_mean	X		X	X	edge:labB_mean	edge:labB_mean	edge:labB_mean
edge:labA_sd	X	X	X	X	X	X	X
edge:labA_skew	X		L:sd_dist	X	edge:labB_skew	edge:labB_skew	edge:labB_skew
edge:labB_mean	X	X	L:sd_dist	**O**	cent:labB_mean	cent:labB_mean	X
edge:labB_sd	X	X	X	X	X edge:labA_sd	X	X
edge:labB_skew	X	X	X	X	X	X	X
edge:hu2	X		L:mean_dist	edge:hu4	edge:hu4	edge:hu4	edge:hu4
edge:hu4	X		L:mean_dist	X	cent:dist	cent:dist	cent:dist

Taulukko 11: Lajeittaiset luokittelutarkkuudet (%) sekä keskiarvo rakenteeltaan rajoittamattomille malleille, joiden piirremuuttujat ovat 20-luokkaisia.

	Päivän-kakkara	Voikukka	Leinikki	Valko-Valkovuokko	Orvokki	Keskiarvo
Naive	88,8	97,5	91,3	93,8	97,5	93,8
K2-naive	88,8	97,5	95,0	95,0	97,5	94,8
BIC-naive	68,8	76,3	68,8	67,5	85,0	73,3
K2-empty	90,0	96,3	92,5	91,3	92,5	92,5
K2b-empty	90,0	100,0	91,3	95,0	98,8	95,0
Naive-K2b	88,8	98,8	93,8	95,0	98,8	95,0

Rakenteeltaan rajoittamattomien mallien sopivuutta aineistoon tarkasteltiin jälleen *leave one out* -ristiinvalidoinnilla. Taulukossa 11 on kuvattu lajeittaiset luokittelutarkkuudet rakenteeltaan rajoittamattomille malleille, joiden piirremuuttujat ovat 20-luokkaisia. Vertailun vuoksi mukana on myös vastaava naiivi Bayes-malli, Naive, sekä naiivi-malli, Naive-K2b, jossa mukana ovat samat muuttujat kuin K2b-empty-mallissa. Mallikohtaiset sekaannusmatriisit on esitetty liitteessä C. Eniten mallien joukosta poikkeaa BIC-naive-malli, jonka luokittelutulokset ovat selvästi huonompia muihin malleihin verrattuna. BIC-kriteeri poistaa mallista liikaa piirteitä ja vaikuttaa siten rankaisevan liikaa verkon solmujen määrästä. Parhaimmat luokittelutulokset antavat mallit K2b-empty ja Naive-K2b, joista on poistettu ison terälehtiluokan muuttujat, deskriptorimuuttujat d6 ja d7 sekä momenttimuuttuja S1:hu1. K2-naive-malli antaa lähes yhtä hyviä tuloksia. Tässä mallissa K2b-malleista poistetut muuttujat ovat kerääntyneet yhteen ison terälehtiluokan muuttujien L:mean_angle ja L:sd_angle alle eivätkä siten enää vaikuta lajiluokkaan, mikä parantanee luokittelutuloksia vastaavaan naiiviin malliin verrattuna. K2-empty-mallin luokittelutulokset ovat toiseksi huonoimmat. Ilmeisesti malliin jää liian vähän suoraan lajiluokkaan vaikuttavia piirteitä; sen sijaan piirteet muotoutuvat mutkikkaaseen hierarkkiseen rakenteeseen suuren terälehtiluokan piirteiden alle.

5-luokkaisten piirremuuttujien malleissa muodostuu yhteyksiä pitkälti samojen piirrevektoreiden välille kuin K2b-empty-mallissa (taulukko 10). K2b-empty-mallista poiketen yhteyksiä muodostuu kuitenkin myös kukan väriä kuvaavien muuttujien välille. Yhteyksien lisääntymiseen vaikuttanee se, että luokkamäärän pienentäminen vähentää yksittäisen piirrevektorin jakauman estimointiin tarvittavien parametrien määrää, jolloin mukaan mahtuu parametrien määrästä rankaisevan sopivuuskriteerin näkökulmasta enemmän muuttujien välisiä yhteyksiä ja näitä mallintavia parametreja. Erityisesti tämä on nähtävissä BIC-sopivuuskriteerin malleissa. 20-luokkaisten piirremuuttujien mallissa BIC-naive mukaan tulee vain 4 piirremuuttujaa, kun taas 5-luokkaisen piirremuuttujan mallissa BICb-2C5 mukana on edelleen suurin osa piirremuuttujista. Lisäksi BICb-2C5-mallissa piirteiden välille muodostuu yhteyksiä, jotka ovat hyvin pitkälti samoja kuin vastaavissa K2-kriteerin malleissa. Piirrevektorin luokkajaon harventaminen voi vaikuttaa yhteyksien lisääntymiseen myös siten, että harvaluokkaiset piirremuuttujat muistuttavat todennäköisemmin toisiaan kuin tiheäluokkaiset piirremuuttujat. Harva jaottelu ei välttämättä riitä luomaan muuttujien välille eroa, joka tiheässä jaottelussa on nähtävissä. Mitä harvempi jako tehdään, sitä enemmän informaatiota menetetään.

K2b-2C5- ja BICb-2C5-malleissa kullekin piirremuuttujalle sallittiin korkeintaan kaksi vanhempaa. Taulukosta 10 voidaan kuitenkin havaita, että BIC-kriteerin mallissa kullakin piirteellä on korkeintaan yksi vanhempi. K2-kriteerin mallissa joillakin piirteillä on kaksi vanhempaa. Taulukosta 12, jossa on esitetty 5-luokkaisten piirremuuttujien mallien luokittelutarkkuudet, voidaan edelleen havaita, että monivanhempinen K2b-2C5-malli tuottaa huonompia luokittelutuloksia kuin yksivanhempinen K2b-1C5-malli. Kaikkien lajien luokittelutarkkuudet huononevat. Kahden vanhemman salliminen tuottanee liian mutkikkaan rakenteen ja ylisovitetun mallin.

Taulukkoon 12 on rajoittamattomien mallien lisäksi laskettu luokittelutulokset kahdesta naiivista mallista. Naive-K2b-C5 on K2b-1C5-mallin pohjalta rakennettu naiivi malli ja sisältää siten kaikki muut piirremuuttujat paitsi suuren terälehtiluokan piirteet. Naive-BICb-C5 on BICb-2C5-mallin pohjalta rakennettu malli, joten mallista on poistettu suurten terälehtiluokkien piirteiden lisäksi deskriptorit d6 ja d7 sekä kukan keskiosan momenttipiirteet S1:hu1,

cent:hu2 ja cent:hu4. Molemmissa tapauksissa naiivit mallit tuottavat prosenttiyksikön verran huonommat luokittelutulokset kuin hierarkkiset vastineensa.

Taulukko 12: Lajeittaiset luokittelutarkkuudet (%) sekä keskiarvo rakenteeltaan rajoittamattomille malleille, joiden piirremuuttujat ovat 5-luokkaisia.

	Päivän-kakkara	Voikukka	Leinikki	Valko-vuokko	Orvokki	Keskiarvo
K2b-2C5	86,3	95,0	91,3	92,5	90,0	91,0
K2b-1C5	90,0	98,8	95,0	95,0	96,3	95,0
Naive-K2b-C5	88,8	97,5	91,3	95,0	97,5	94,0
BICb-2C5	90,0	98,8	97,5	92,5	95,0	94,8
Naive-BICb-C5	90,0	96,3	95,0	92,5	95,0	93,8

4.6.3 Havaintoja menetelmän ominaisuuksista ja käyttökelpoisuudesta

Taulukoista 11 ja 12 havaitaan, että 5-luokkaisten piirremuuttujien mallien luokittelutarkkuudet eivät juurikaan eroa parhaiden 20-luokkaisten piirremuuttujien mallien luokittelutarkkuuksista. Tässä aineistossa 5 luokkaa näyttäisi siten olevan riittävä luokkamäärä. Myös eri sopivuuskriteerien tuottamat mallit muistuttavat enemmän toisiaan 5- kuin 20-luokkaisilla piirremuuttujilla. BIC-kriteeri ei pysty estimoimaan 20-luokkaisille piirremuuttujille järkevää rakennetta, vaan jättää lähes kaikki muuttujat pois mallista. Taulukossa 13 on esitetty eri mallien sopivuuskriteerit. Arvoltaan suurempi sopivuuskriteeri on parempi. 20-luokkaisten piirremuuttujien malleissa BIC-sopivuuskriteeri on naiiveissa malleissa aina suurempi kuin vastaavissa hierarkkisia rakenteita sisältävissä malleissa. 5-luokkaisilla piirremuuttujilla BIC-sopivuuskriteeri on naiiveissa malleissa sen sijaan pienempi kuin vastaavissa hierarkkisen rakenteen sisältävissä malleissa, joissa kullakin piirremuuttujalla on korkeintaan yksi vanhempi. Tämäkin kertoo osaltaan siitä, että 20 luokkaa piirremuuttujassa lienee liian suuri määrä tämänkokoiseen aineistoon.

Luokittelutulokset tukevat myös sitä, että alkuperäisestä piirremuuttujien joukosta voitaisiin hyvin vähentää joitakin muuttujia. Esimerkiksi naiivi-malli, josta puuttuvat ison terälehti-

Taulukko 13: Rakenteeltaan rajoittamattomien mallien sopivuuskriteerit. Arvoltaan suurempi sopivuuskriteeri on parempi. Sopivuuskriteerejä ei kuitenkaan voi suoraan verrata keskenään, koska muuttujien määrä malleissa vaihtelee. Mallien nimet on selitetty luvun 4.6.2 alussa.

	K2	BIC
Naive	-38775,2	-44392,1
K2-naive	-38342,5	47367,2
BIC-naive	-4345,9	-4864,9
K2-empty	-37861,1	-58839,1
K2b-empty	-31045,2	-38845,0
Naive-K2b	-31167,5	-35581,6
K2b-2C5	-15939,3	-17920,0
K2b-1C5	-16051,1	-16671,4
Naive-K2b-C5	-16473,3	-17085,7
BICb-2C5	-13318,6	-13808,5
Naive-BICb-C5	-13716,4	-14207,2

luokan piirteet, keskiosan avainpisteiden momenttipiirre ja kaksi deskriptoripiirrettä, antaa paremman luokittelutuloksen kuin alkuperäinen, kaikki piirrevektorit sisältävä naiivi Bayes-malli. 5-luokkaisten piirremuuttujien naiivit mallit, Naive-K2b-C5 ja Naive-BICb-C5, tuottavat lähes samat luokittelutulokset, vaikka Naive-BICb-C5-malli sisältää jopa viisi piirremuuttujaa vähemmän (taulukot 10 ja 12). Edelleen rakenteeltaan rajoittamattomat mallit, joissa osa piirremuuttujista muodostaa hierarkkisen rakenteen, tuottavat keskimäärin parempia luokittelutuloksia kuin naiivit vastineensa — edellyttäen, että hierarkkiset rakenteet eivät monimutkaistu liikaa. Tämäkin kertoo siitä, että malleissa on mukana luokittelun kannalta turhia muuttujia tai vastaavasti muuttujia, joilla on osin sama informaatiosisältö. Luokkasolmuun vaikuttavat vain piirteet, jotka kuuluvat luokkasolmun Markovin peitteeseen (Friedman, Geiger ja Goldszmidt 1997). Esimerkiksi lapsisolmuiksi jonkin toisen piirremuuttujan alle asettuvat piirteet vaikuttavat luokittelutulokseen vain, mikäli vanhempisolmusta sattuu puuttumaan havaintoja. Tällaisessa tapauksessa mallirakenteen kannalta katsot-

tuna lapsisolmut ovat täysin redundantteja vanhempisolmujen suhteen. Luokkasolmun Markovin peitettä voisikin varauksella käyttää valitsemaan malliin sopivia piirteitä (Friedman, Geiger ja Goldszmidt 1997; Cheng ja Greiner 1999). Toinen vaihtoehto voisi olla yhdistää korreloivia, mitattuja piirteitä ylemmän tason piilomuuttujiksi, jotka eivät ole suoraan datasta havaittavissa. Tässä vaihtoehdossa molemmat piirteet olisivat yhä mallissa mukana eikä kummankaan informaatiosisältö korvaisi täysin toista muuttujaa, kuten käy valittaessa piirteet Markovin peitettä hyödyntäen.

Kuten taulukosta 10 käy ilmi, eri sopivuuskriteerit tuottivat erilaiset rakenteet. Mallien rakenteet olisi voitu muodostaa myös muutoin kuin tarkastelemalla mallin sopivuutta aineistoon sopivuuskriteereillä. Cheng ja Greiner (1999) esimerkiksi hyödynsivät artikkelissaan muuttujien välisiä riippumattomuusmittoja, kuten yhteistä informaatiota. Friedman, Geiger ja Goldszmidt (1997) kritisoivat artikkelissaan BIC-sopivuuskriteeriä siitä, että se painottaa liikaa mallin sopivuutta piirremuuttujien yhteistodennäköisyysjakaumaan luokan piirteillä ehdollistetun todennäköisyysjakauman kustannuksella. Erityisesti tämä tulee esiin tilanteissa, joissa piirteitä on paljon. Tällöin suhteellisen suuri virhe luokan ehdollisessa todennäköisyysjakaumassa saattaa jäädä huomiotta. Vertaillessaan BIC-kriteerillä estimoidun yleisen verkon tuloksia naiivin Bayes-verkon tuloksiin Friedman, Geiger ja Goldszmidt (1997) havaitsivatkin naiivin verkon olevan parempi luokittelija juuri niissä tapauksissa, joissa piirteitä oli paljon.

Tässä tutkimuksessa rakenteeltaan rajoittamattomien verkkojen luokkasolmua kohdeltiin tasaveroisena muiden solmujen kanssa. Teoriassa luokkasolmu olisi voitu myös valita aina juurisolmuksi (Cheng ja Greiner 1999), mutta pgmpy-kirjaston nykyinen rakenne-estimointialgoritmi ei mahdollistanut tätä. Yleistä Bayes-luokittelijaa voitaisiin rajoittaa myös kohdetiedon perusteella esimerkiksi estämällä epäluonnolliset yhteydet piirteiden välillä, tai hyödyntämällä tiedettyjä tai oletettuja syy-seuraussuhteita piirteiden välillä (Cheng ja Greiner 1999).

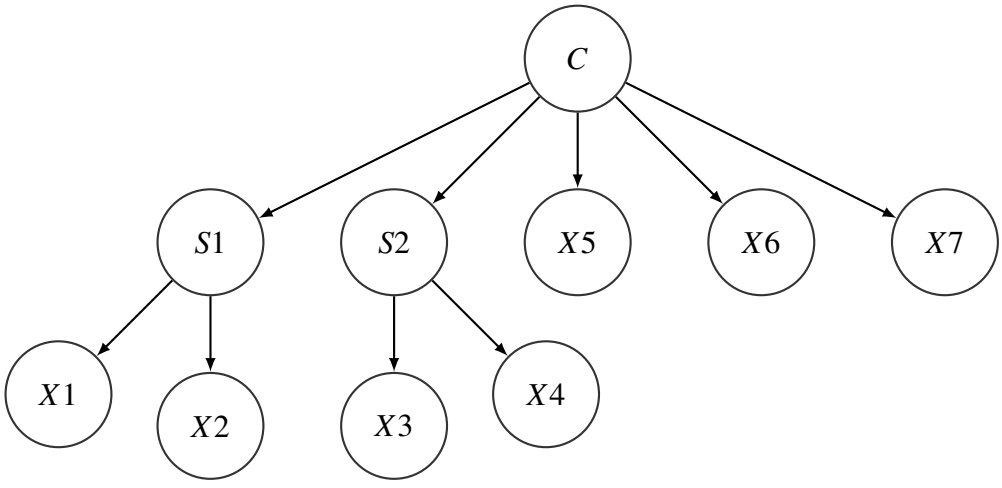
Kun tarkastellaan sekä luokittelutuloksia että sopivuuskriteerejä, sopiva malli rakenteeltaan rajoittamattomien mallien joukossa tutkimuksen aineistolle voisi olla BICb-2C5-malli. Mallin piirremuuttujat ovat 5-luokkaisia, ja mallista puuttuvat suuren terälehtiluokan piirteet, deskriptorit d6 ja d7 sekä kukan keskusosan momenttimuuttujat S1:hu1, cent:hu2 ja cent:hu4.

Mallista poisjääneiden muuttujien vaikutus lajintunnistukseen on vähäinen myös taulukon 6 perusteella. Toisaalta lajeittain luokittelutarkkuuksia tarkasteltaessa BICb-2C5-mallia tasaisemman tuloksen antaa K2b-1C5-malli, josta puuttuvat vain suurten terälehtiluokkien piirteet.

4.7 Hierarkkiset mallit koostemuuttujilla

4.7.1 Mallin kuvaus ja koostemuuttujien muodostaminen

Eräs tapa ottaa huomioon piirteiden välisiä riippuvuuksia ja muodostaa hierarkkinen verkko on koostaa mitattuja, toisistaan riippuvaisia piirteitä uusiksi muuttujiksi. Mallissa koostemuuttujat ovat suoraan vanhempisolmuna toimivan luokkamuuttujan alla ja piirremuuttujat puolestaan koostemuuttujien alla. Malli voi sisältää myös suoraan luokkasolmun alla olevia piirremuuttujia. Esimerkki koostemuuttujia sisältävästä mallista on kuviossa 28. Koostemuuttujat ovat käyttökelpoisia erityisesti silloin, kun koostettavat piirteet sisältävät osin samaa informaatiota, mutta ne eivät kuitenkaan ole täysin korvattavissa toinen toisillaan. Koostemuuttuja sisältää informaatiota kaikista yhdistämistään piirteistä. Siten kaikki nämä piirteet vaikuttavat edelleen myös luokkamuuttujaan toisin kuin mallissa, jossa piirremuuttuja asettuu toisen piirteen lapseksi kadottaen yhteyden luokkamuuttujaan.



Kuvio 28: Esimerkki koostemuuttujia sisältävästä Bayes-verkosta. Koostemuuttujat on merkitty *S*-kirjaimella ja havaitut piirteet *X*-kirjaimella. *C* on luokkamuuttuja.

Piirremuuttujien riippuvuutta toisistaan tarkasteltiin d_{χ^2} -poikkeavuusmitalla (kaava 3.6) sekä Pearsonin ja Spearmanin korrelaatioilla[17]. Korrelaatioiden avulla tarkasteltiin, oliko riippuvuus lineaarista. Spearmanin korrelaatio laskettiin Pearsonin korrelaation lisäksi, koska muuttujat olivat diskreettejä. Järjestyslukuihin perustuvana Spearmanin korrelaatio antaa Pearsonin korrelaatiota luotettavampia tuloksia myös tilanteessa, jossa joukossa on poikkeavia havaintoja. Taulukossa 14 on kuvattu 15 ensimmäisen piirremuttujan väliset riippuvuusluvut, kun muuttujat on järjestetty Pearsonin korrelaatiokertoimen itseisarvon perusteella laskevaan järjestykseen. Taulukkoa 14 tarkastelemalla havaitaan, että järjestyksessä ensimmäisten piirteiden välillä on vahvaa (lineaarista) riippuvuutta, joten muuttujien yhdistäminen on perusteltua.

Muuttujia voidaan yhdistää monella tapaa. Eräs yksinkertainen menetelmä on muodostaa muuttujista summamuuttuja laskemalla yhteen yhdistettävien muuttujien arvot. Koska mitattujen piirteiden riippuvuus oli korrelaatioiden perusteella luonteeltaan lineaarista, muodostettiin koostemuuttujat myös tässä tapauksessa piirremuuttujia yhteen summaamalla. Piirremuuttujia yhdistettiin taulukossa 14 kuvatussa järjestyksessä ja mukaan otettiin sellaiset muuttujaparit, joiden Pearsonin korrelaation itseisarvo oli vähintään 0,7. Muuttujien valinnassa päädyttiin käyttämään Pearsonin korrelaatiokerrointa, koska Pearsonin kerroin korreloi hieman voimakkaammin d_{χ^2} -poikkeavuusmitan kanssa kuin Spearmanin korrelaatiokerroin, $r_P = 0,668$ vs. $r_S = 0,655$. Toisaalta Pearsonin ja Spearmanin korrelaatioiden keskinäinen korrelaatio oli voimakas, $r = 0,975$, joten Pearsonin korrelaatiota voidaan pitää suhteellisen luotettavana, vaikka muuttujien jakaumia ei olekaan sen kummemmin tarkasteltu.

Summamuuttujia kokeiltiin muodostaa kahdella eri tavalla. Ensimmäisessä tavassa piirremuuttujien arvot laskettiin suoraan yhteen ja saatu summa luokiteltiin halutulle välille. Toisessa tavassa kutakin summattavaa piirremuuttujaa painotettiin piirteen ja lajiluokan välille lasketulla poikkeavuusmitan arvolla. Näin koostemuuttujaan vaikutti eniten se piirre, jolla oli suurin vaikutus myös lajiluokkaan. Summamuuttujan luokkien määräksi valittiin 5. Syitä luokkien suhteellisen pieneen määrään oli monia. Usean uuden moniluokkaisen muuttujan lisääminen malliin aiheuttaa jo sinällään mallin monimutkaistumista. Lisäksi 5-luokkaisten koostemuuttujien mallissa on käytettävissä huomattavasti enemmän havaintoja estimoitavaa

17. Ks. scipy-dokumentaatio: mstats.pearsonr/spearmanr

parametria kohti kuin 20-luokkaisten koostemuuttujien mallissa, mikä parantaa estimaattien luotettavuutta. Edelleen rakenteeltaan rajoittamattomien mallien tuloksista havaittiin, että luokittelutarkkuudet olivat samaa luokkaa sekä 5- että 20-luokkaisten piirremuuttujien malleissa.

Taulukko 14: Piirremuuttujien välisiä parittaisia riippuvuuksia. Taulukko on järjestetty laskevaan järjestykseen Pearsonin korrelaatiokertoimen r_P itseisarvon mukaan ja se sisältää 15 ensimmäistä muuttujaparia. Yhdistetyt muuttujat on merkitty mallisarakkeisiin numeroin. Samalla numerolla merkatut muuttujaparit on yhdistetty yhdeksi koostemuuttujaksi. *A* ja *B* kuvaavat eri mallityyppejä.

mja1	mja2	d_{χ^2}	r_P	r_S	mallit A	mallit B
L:sd_dist	L:sd_angle	1839,2	0,932	0,875	1	
S:mean_angle	S:sd_angle	1285,4	0,897	0,914	2	1
L:mean_dist	L:mean_angle	1232,8	0,823	0,773	3	
cent:labB_mean	cent:labB_skew	919,3	-0,806	-0,744	4	2
edge:labB_mean	edge:labB_skew	745,4	-0,801	-0,787	5	3
cent:dist	edge:hu4	685,2	0,737	0,748	6	4
S:mean_dist	S:sd_dist	812,5	0,737	0,762	7	5
S:mean_angle	S2:hu1	726,3	0,721	0,757		1
cent:hu2	cent:hu4	882,1	0,717	0,699		6
edge:labA_sd	edge:labB_sd	763,3	0,708	0,795		7
cent:labB_sd	edge:labB_mean	720,3	-0,695	-0,671		
S:sd_angle	S2:hu1	741,5	0,690	0,768		1
S:mean_dist	S:sd_angle	546,6	0,667	0,645		
S:mean_dist	S:mean_angle	606,2	0,661	0,601		
cent:labB_mean	edge:labB_mean	697,2	0,650	0,699		

Koostemuuttujia sisältävän mallin piirremuuttujat ovat samat kuin aiemmissa malleissa. Muuttujat on kuvattu luvussa 4.4.2. Osassa koostemuuttujamalleja piirteiden määrää vähennettiin vastaamaan piirteitä, jotka olivat mukana erilaisissa rakenteeltaan rajoittamattomissa malleissa (ks. luku 4.6.2). Suurimmassa osassa koostemuuttujamalleja piirremuuttujat ovat 20-luokkaisia. Myös piirteiden väliset riippuvuustarkastelut toteutettiin 20-luokkaisille piirremuuttujille. Osassa koostemuuttujamalleja kokeiltiin kuitenkin 5-luokkaisia piirremuuttujia, jotta pystyttiin vertailemaan tuloksia rakenteeltaan rajoittamattomien, 5-luokkaisten piirremuuttujien mallien tuloksiin.

4.7.2 Mallien rakenteet ja luokittelutulokset

Muistin loppuminen ennustevaiheessa rajoitti jossain määrin summamuuttujien muodostamista. Mallin sisältäessä kaikki piirremuuttujat korrelaatiorajaksi valikoitui 0,725. Summamuuttujia muodostui tällöin 7. Tätä suuremmilla summamuuttujien määrillä muisti loppui kesken sekä viestinvälitys- että muuttujien eliminointi -ennustealgoritmeilla. Muistin loppumiseen lienee vaikuttanut koostemuuttujien marginaalisummien kombinaatioiden suuri määrä ja laskennan aikana tehtävät välitallennukset. Koostemuuttujien ryhmittyessä luokkasolmun alle luokkasolmuun vaikutti yhtäaikaa paljon muuttujia, joista ei ollut havaintoja. Malleissa, joissa ei ollut enää mukana suuren terälehtiluokan muuttujia, kaikki korrelaation rajaarvon ylittävät koostemuuttujat saatiin kuitenkin mukaan, joten kovin suureksi ongelmaksi muistin loppuminen ei muodostunut.

Taulukon 14 sarakkeisiin *mallit A* ja *mallit B* on merkitty numeroin ne muuttujat, jotka on yhdistetty kulloisessakin mallityypissä koostemuuttujiksi. Malleissa *A* mukana ovat olleet suuren terälehtiluokan piirteet, mutta malleista *B* ne on poistettu. Muilla poistetuilla piirteillä ei ollut matalien korrelaatioiden vuoksi vaikutusta koostemuuttujien muodostumiseen. Samaan summamuuttujaan koostetut piirteet on merkitty samalla numerolla. Useimmiten koostemuuttujaan on yhdistetty vain kaksi muuttujaa. Malleissa *B* yhteen koostemuuttujaan on kuitenkin yhdistetty kolme piirrettä: S:mean_angle, S:sd_angle ja S2:hu1. Tarkkaan ottaen muuttujaparin (Ssd_angle, S2:hu1) korrelaatiokerroin on alle 0,7, mutta tämäkin muuttujapari päädyttiin ottamaan mukaan, jotta voitiin koostaa kolmen piirteen summamuuttuja. Sisällöllisesti koostemuuttujat vaikuttavat järkeviltä. Esimerkiksi kolmen muuttujan yh-

disteen kaikki piirteet kuvastavat kukan reunaosan pienien avainpisteiden sijaintia toisiinsa nähden. Muut koostemuuttujat yhdistävät muun muassa reuna- tai keskiosan väriä kuvaavia piirteitä tai keskusosan momenttipiirteitä. Koostemuuttujien yhdistämät piirteet ovat myös suurelta osin niitä samoja piirteitä, jotka asettuivat hierarkkiseksi rakenteeksi rakenteeltaan rajoittamattomissa malleissa.

Koostemuuttujia sisältävien mallien sopivuutta aineistoon tarkasteltiin jälleen *leave one out* -ristiinvalidoinnilla. Taulukossa 15 on esitetty koostemuuttujamallien luokittelutarkkuudet sekä lajeittainen keskiarvo. Mallikohtaiset sekaannusmatriisit on kuvattu liitteessä D. Mallien nimilyhenteet ovat seuraavat: *A* tarkoittaa, että mukana ovat kaikki piirteet. Malleista *B1* on poistettu ison terälehtiluokan piirteet, deskriptorimuuttujat d6 ja d7 sekä keskiosan pienen skaalan avainpisteiden momenttimuuttuja. *B2*-mallista on poistettu vain ison terälehtiluokan piirteet. Pieni *w*-kirjain tarkoittaa, että piirteitä on painotettu vaikutuksella lajiluokkaan koostemuuttujaa laskettaessa. Jos *w* puuttuu, piirteet ovat painottamattomia. Mallien *B* loppuliite *C* kertoo piirremuuttujien luokkien määrän: *C20*-malleissa se on 20 ja *C5*-malleissa 5. B1w-C20-malli vastaa siten piirremuuttujiltaan rakenteeltaan rajoittamattomien mallien K2b-empty-mallia, ja B2w-C5-malli vastaa rajoittamattomien mallien K2b-C5-malleja.

Taulukosta 15 havaitaan, että painotettujen koostemuuttujamallien luokittelutulokset ovat keskimäärin selvästi parempia kuin painottamattomien koostemuuttujamallien. Ero on suurempi malleissa, joissa piirteiden määrää on vähennetty alkuperäisestä. Kun kaikki piirteet sisältävää Aw-mallia vertaa vastaavaan naiiviin malliin ja K2-sopivuuskriteerin tuottamaan, osin hierarkkiseen K2-naive-malliin (taulukko 11), havaitaan, että luokittelutarkkuus on samaa luokkaa naiivin mallin kanssa, mutta K2-naive-mallin tarkkuus on prosenttiyksikön verran parempi. B1w-C20-mallin tulokset ovat samansuuntaisia. Keskimääräinen luokittelutarkkuus on täsmälleen sama kuin vastaavassa K2-kriteerin tuottamassa mallissa K2b-empty tai naiivissa vastineessa Naive-K2b. 5-luokkaisten piirremuuttujien mallien vertailu tuottaa niinikään vastaavan tuloksen. B2w-C5-mallin luokittelutarkkuus on sama kuin naiivin vastineen Naive-K2b-C5. Prosenttiyksikön paremman luokittelutuloksen tuottaa vastaava K2-malli K2b-1C5, jossa kullekin piirteelle on sallittu korkeintaan yksi vanhempi. Lajeittai-

set luokittelutulokset kuitenkin vaihtelevat. Päivänkakkara tuntuisi hyötyvän koostemuuttujien muodostamisesta, mutta orvokille käy päinvastoin erityisesti piirremuuttujien ollessa 5-luokkaisia.

Taulukko 15: Lajeittaiset luokittelutarkkuudet (%) sekä keskiarvo hierarkkisille koostemuuttujamalleille.

	Päivänkakkara	Voikukka	Leinikki	Valkovuokko	Orvokki	Keskiarvo
Aw	88,8	98,8	91,3	93,8	96,3	93,8
A	90,0	93,8	81,3	91,3	88,8	89,0
B1w-C20	91,3	97,5	92,5	96,3	97,5	95,0
B1-C20	87,5	97,5	80,0	88,8	82,5	87,3
B2w-C5	92,5	97,5	95,0	92,5	92,5	94,0
B1w-C5	93,8	97,5	97,5	93,8	92,5	95,0

Vertailun vuoksi lajiluokkaa kokeiltiin ennustaa myös käyttämällä suoraan koostemuuttujia. Tällöin hierarkiataso häviää ja mallista muodostuu naiivi Bayes-malli, joka koostuu alkuperäisistä piirremuuttujista ja toisista piirremuuttujista rakentuneista koostemuuttujista. Näissä malleissa koostemuuttujat laskettiin painotetuista piirremuuttujista. Taulukkoon 16 on koottu koostemuuttujamallien luokittelutarkkuudet, kun ennustetta laskettaessa on käytetty suoraan koostemuuttujia. Suurimmalla osalla malleista suoraan summamuuttujista laskettu ennuste antaa keskimäärin jonkin verran huonompia luokittelutuloksia verrattuna piirremuuttujien käyttämiseen. Erot ovat kuitenkin hyvin pieniä. 5-luokkaisten piirremuuttujien B2w-C5-mallissa suoraan summamuuttujista laskettu luokittelutarkkuus on jopa hieman parempi.

4.7.3 Havaintoja menetelmän ominaisuuksista ja käyttökelpoisuudesta

Koostemuuttujia sisältävät mallit antoivat osapuilleen samanlaisia luokittelutuloksia kuin naiivit ja rakenteeltaan rajoittamattomat Bayes-mallit. Luokittelutarkkuudet alkavat kuitenkin olla valitussa aineistossa jo niin hyviä, että mallien vertailu on vaikeaa; kyse on muutaman kuvan luokittelusta toiseen luokkaan, mistä ei voi tehdä systemaattista tulkintaa. Koostemuuttujamallien ongelmana on, että ne tuovat lisää muuttujia malliin ja sitä kautta moni-

Taulukko 16: Lajeittaiset luokittelutarkkuudet (%) sekä keskiarvo koostemuuttujamalleille, kun ennustamisessa on käytetty suoraan summamuuttujia.

	Päivänkakara	Voikukka	Leinikki	Valkovuokko	Orvokki	Keskiarvo
Aw	90,0	98,8	91,3	90,0	95,0	93,0
B1w-C20	92,5	95,0	90,0	92,5	96,3	93,3
B2w-C5	91,3	97,5	93,8	93,8	95,0	94,3
B1w-C5	92,5	97,5	95,0	93,8	95,0	94,8

mutkaistavat mallia. Toisaalta etuna siihen nähden, että hierarkkiset rakenteet muodostuvat mitattujen piirteiden välille, on se, että kaikki koostemuuttujaan yhdistetyt havaitut piirteet vaikuttavat koostemuuttujan kautta lajiluokkaan.

Koostemuuttujamalleihin liittyvä oleellinen kysymys on, kuinka koostemuuttujat tulisi muodostaa. Tässä tutkimuksessa koostemuuttujat rakennettiin yksinkertaisesti laskemalla suora tai painotettu summa korreloivista piirremuuttujista. Painona käytettiin piirteen ja lajiluokan välille laskettua poikkeavuusmitan arvoa. Paino olisi voinut olla esimerkiksi myös lajiluokan ja piirteen yhteinen informaatio tai jokin muu suure, joka mittaa piirteen vaikutusta lajiluokkaan. Todennäköisesti samaa ominaisuutta mittaavat erilaiset suureet olisivat kuitenkin antaneet yhteneväisiä tuloksia. Painot olisi voitu myös oppia käyttäen sopivuuskriteeriä tai luokittelutarkkuutta oppimiskriteerinä. Selvästi piirteiden painotus koostemuuttujissa kuitenkin paransi luokittelutulosta.

Koostemuuttujat olisi voitu summaamisen sijaan muodostaa myös esimerkiksi pääkomponenttianalyysilla tai klusteroimalla. Voisi olla mahdollista muodostaa esimerkiksi väripiirteistä eri värejä kuvaavia klustereita, jotka puolestaan kuvaisivat millä todennäköisyydellä tietyn lajin kukka on vaikkapa punainen. Koostemuuttujilla olisi siten tässä tapauksessa myös selkeä semanttinen merkitys. Tällä tavoin voitaisiin rakentaa Bayes-verkko, joka toimisi myös ihmismielelle sisällöllisesti käsitettävänä lajintunnistusoppaana. Jo tämänkin tutkimuksen summamuuttujille voidaan asettaa sisällöllisiä merkityksiä: esimerkiksi kolmen piirteen yhdiste (S:mean_angle, S:sd_angle, S2:hu1) kuvaa kukan terälehtien kokoa ja määrää.

Tutkimuksessa kokeiltiin ennustaa lajiluokkaa myös suoraan summamuuttujilla koostettujen piirteiden sijaan. Ennustaminen suoraan summamuuttujista antoi kuitenkin yhtä mallia lukuunottamatta hieman huonompia luokittelutuloksia, vaikka tarkastelussa käytettiinkin aina painotettuja summia. Yksinkertainen suora summa on helppo laskea myös testiaineiston piirremuuttujista. Mutta jo painotetun summan laskenta aiheuttaa ongelmia: testihavainnolle koostemuuttujia laskettaessa tulisi olla tieto siitä, miten paino muodostetaan. Lisäksi testiaineistossa ei välttämättä ole havaittu kaikkia piirrevektoreita. Havaintojen puuttuminen aiheuttaisi ongelmia koostemuuttujien muodostamiseen. Kun ennustamiseen käytetään havaittuja piirteitä, vain koostemuuttujien posterioritodennäköisyys muuttuu tehtyjen havaintojen myötä.

5 Yhteenveto

Tässä tutkimuksessa rakennettiin Bayes-verkkoihin pohjautuva luokittelija, joka pystyi varsin hyvällä tarkkuudella ennustamaan oikein kuvassa olevan kukkivan kasvin lajin. Luokittelijan rakentamiseen sisältyi sekä kuvadatan piirteistäminen tilastollisiksi muuttujiksi että luokittelumallin rakentaminen. Tutkimuksessa testattiin useita erityyppisiä Bayes-verkkoja, jotka erosivat toisistaan eritoten kyvyssä huomioida piirremuuttujien välisiä tilastollisia riippuvuuksia.

Tutkimuksen alkuperäisenä tavoitteena oli mallintaa kukkakuvien data osapohjaisella mallilla. Se osoittautui kuitenkin liian vaativaksi maisteritason lopputyöhön nähden. Oli myös hankala määrittää, miten kukan osajako olisi pitänyt tehdä. Keskiosan lisäksi kukasta on löydettävissä reunaosa, joka muodostuu jopa useasta kymmenestä terälehdestä. Jos terälehtiä ajattelee osina, ulkomuoto-ominaisuuksiltaan enemmän tai vähemmän samanlaisia osia voisi siten olla kymmeniä. Piirrevektoreihin pyrittiin kuitenkin sisällyttämään jonkinlaista spatiaalista informaatiota jakamalla kukka reuna- ja keskusosaan ja laskemalle joitakin piirrevektoreita kummallekin osalle erikseen. Etäisyystietoa kohteen keskipisteestä hyödynnettiin myös avainpisteiden luokittelussa, ja lisäksi luokittelumallissa oli mukana piirrevektoreita, jotka kuvasivat terälehtien sijaintia toisiinsa nähden. Luvussa 2.1.3 mainituista ulkomuotomalleista tässä tutkimuksessa käytetty malli vastasi lähinnä globaalia ikkunointimallia, mutta huomioi kuitenkin sitä paremmin spatiaalista informaatiota rotaatioinvarianttiutta vaarantamatta.

On hankala tulkita, mikä tutkimuksen Bayes-malleista on paras, sillä parhaimpien mallien luokittelutarkkuudet erosivat toisistaan alle prosenttiyksikön verran. Parhaimmat mallit antoivat myös varsin hyviä luokittelutuloksia: keskimäärin 95 % kukista luokittui oikeaan lajiluokkaan. Toisin sanoen 80 havaintoyksikön aineistossa keskimäärin neljä kuvaa ennustettiin väärään lajiluokkaan. Tutkimuksessa onnistuttiin siten löytämään piirteitä, jotka kykenivät erottelemaan kasvilajeja toisistaan. Toki on muistettava, että kasvilajiluokkien määrä tutkimuksessa oli vain viisi. Mukaan otetut lajit olivat kuitenkin sellaisia, että niitä ei olisi pystynyt erottamaan toisistaan hyvin pelkän väri-informaation perusteella — luokittelutulosten havaittiinkin keskimäärin paranevan selvästi, kun mukaan otettiin enemmän kukan muotoa

Taulukko 17: Parhaimpien mallien lajeittaiset luokittelutarkkuudet (%) sekä tulosten keskiarvo ja keskihajonta.

Malli	K2b-empty	Naive-K2b	K2b-1C5	B1w-C20	B1w-C5
Päivänkakkara	90,0	88,8	90,0	91,3	93,8
Voikukka	100,0	98,8	98,8	97,5	97,5
Leinikki	91,3	93,8	95	92,5	97,5
Valkovuokko	95,0	95,0	95,0	96,3	93,8
Orvokki	98,8	98,8	96,3	97,5	92,5
Keskiarvo	95,0	95,0	95,0	95,0	95,0
Keskihajonta	4,4	4,1	3,2	2,9	2,3

kuvaavia piirteitä. Taulukosta 6 voidaan havaita, että parhaiten lajeja erottelivat toisistaan kukan reunaosan värimuuttujat ja pienien terälehtien luokan pisteiden väliset kulmamuuttujat. Rakenteeltaan rajoittamattomien mallien tuloksia analysoitaessa puolestaan havaittiin, että malleista voitiin poistaa joitakin taulukossa 6 loppupuolella sijaitsevia muuttujia tuloksia huonontamatta.

Taulukossa 17 on kuvattu, miten tasaisesti luokittelutarkkuus jakautuu lajeittain parhaissa malleissa. Tuloksista voidaan havaita, että tasaisimman luokittelutuloksen tuotti koostemuuttujamallien B1w-C5-malli. Mallin piirremuuttujat ovat 5-luokkaisia ja mallista on poistettu isojen terälehtien luokan piirteet, deskriptorimuuttujat d6 ja d7 sekä keskiosan pienen skaalan avainpisteiden momenttimuuttuja. Varovasti tulkiten näyttäisi siltä, että 5-luokkaisten piirremuuttujien mallit antavat hieman tasaisempia luokittelutuloksia kuin 20-luokkaisten muuttujien mallit. Lisävalaistusta mallien keskinäiseen paremmuuteen olisi voinut tuoda jonkin muun mittarin käyttäminen luokittelutarkkuuden ohella. Lucas (2002) hyödynsi artikkelissaan luokkamuuttujan posterioritodennäköisyyden entropiaa (ks. luku 4.5.4). Sekään ei tosin suoraan kerro, onko jonkin muun yksittäisen luokan todennäköisyys lähellä ennustetun luokan todennäköisyyttä. Laskemalla posterioritodennäköisyydet kaikille lajiluokille saataisiin kattava kuva tuloksista. Useiden kymmenien lajiluokkien tapauksessa suora posterioritodennäköisyyksien vertailu olisi kuitenkin jo hankalaa.

Vaikka parhainta mallia ei voidakaan valita, mallien rakenteista voidaan kuitenkin esittää joitakin suuntaviivoja, joista osa antaa aihetta myös jatkotutkimukselle. TAN-malli, jossa kaikilla piirremuuttujilla yhtä lukuunottamatta on luokkamuuttujan lisäksi vanhempana myös jokin toinen piirremuuttuja, oli selvästi liian monimutkainen tähän aineistoon. FAN-mallit, joissa piirrevektorien välillä on TAN-mallia vähemmän yhteyksiä, toimivat paremmin. Rakenteeltaan rajoittamattomien mallien luokittelutuloksia analysoitaessa havaittiin, että ne huononivat, mikäli rakennetta estimoitaessa muuttujalle sallittiin kaksi vanhempaa yhden sijasta. Muutamissa malleissa kokeiltiin myös 5-luokkaisia piirremuuttujia alkuperäisten 20-luokkaisten sijaan. 5-luokkaisten piirremuuttujien mallit antoivat jotakuinkin yhtä hyviä luokittelutuloksia kuin 20-luokkaisten piirteiden mallit, mutta ovat kuitenkin selvästi yksinkertaisempia. 20-luokkaisten piirremuuttujien malleissa, joissa piirremuuttujalla on lajiluokan lisäksi vanhempana toinen piirremuuttuja, jakaumaluokkiin jää niin vähän havaintoja, ettei parametriestimaatteja voida pitää enää luotettavina. Yhteenvetona näistä tuloksista voidaan päätellä, että malli kannattaa toteuttaa mahdollisimman yksinkertaisena. Tällöin todennäköisimmin välttyy myös mallin ylisovittamiselta.

Eräs jatkotutkimuksen aihe olisikin selvittää, miten piirremuuttujien histogrammien luokkajako kannattaisi muodostaa, ja etsiä optimiratkaisu, jossa sekä luokittelutarkkuus että sopivuuskriteeri olisivat mahdollisimman hyviä. Tässä tutkimuksessa histogrammien luokkien määrät olivat lähinnä valistuneita arvioita, ja lisäksi määrä oli sama kaikilla piirremuuttujilla. Eri piirteillä kannattaisi todennäköisesti olla eri määrä jakaumaluokkia riippuen alkuperäisen jatkuvan muuttujan jakauman muodosta. Tulisi tutkia myös sitä, miten histogrammien luokkarajat määrättäisiin — tasavälinen jako ei välttämättä ole paras mahdollinen. Eräs vaihtoehto voisi olla käyttää luokitteluasteikollisten piirremuuttujien sijaan jatkuvia muuttujia. Tällöin kuitenkin käytännössä jouduttaisiin olettamaan, että muuttujat noudattaisivat normaalijakaumaa, mikä ei tämän tutkimuksen muuttujien osalta useinkaan pidä paikkaansa.

Mielenkiintoinen tulos malleja vertailtaessa on myös se, että piirremuuttujien keskinäisiä tilastollisia riippuvuuksia huomioivat mallit eivät antaneet naiivia Bayes-mallia parempia tuloksia. Toki 95 %:n luokittelutarkkuus on jo sen verran hyvä, että kovin paljon tulos ei olisi edes voinut parantua. On myös syytä huomata, että TAN- ja FAN-malleissa piirremuuttujat olivat aina 20-luokkaisia, joten ei tiedetä, olisivatko tulokset muuttuneet, jos olisi käy-

tetty 5-luokkaisia muuttujia. Mallit olisivat tällöin ainakin huomattavasti yksinkertaistuneet sekä kuvanneet luotettavammin piirteiden todennäköisyysjakaumia. Naiivien mallien yksinkertaisuus onkin varmasti yksi syy, miksi ne toimivat hyvin tässä tutkimuksessa. Lisäksi luokkaennuste voi osua oikeaan, vaikka luokkien posterioritodennäköisyydet eivät vastaisikaan todellisia todennäköisyyksiä (Zhang 2004). Empiirisissä vertailututkimuksissa onkin havaittu naiivin Bayes-luokittelijan toimivan monessa tapauksessa hyvin, vaikka piirteiden välillä olisikin voimakasta riippuvuutta (esim. Domingos ja Pazzani 1996). Zhang (2004) tarkasteli artikkelissaan, milloin naiivit luokittelijat toimivat yhtä hyvin kuin piirremuuttujien riippuvuuden huomioon ottavat Bayes-verkot. Jos piirteet riippuvat toisistaan jotakuinkin samalla tavoin kaikissa luokissa, tuottaa naiivi Bayes-luokittelija edelleen hyviä tuloksia, vaikka piirteiden välillä olisikin voimakasta riippuvuutta. Piirteiden parittaiset riippuvuudet voivat myös vaikuttaa saman- tai erisuuntaisesti luokkien todennäköisyyksiin. Kun tarkastellaan kaikkia piirteitä kerralla, erisuuntaiset vaikutukset kumoavat toinen toisensa ja naiivi luokittelija toimii edelleen hyvin.

Koska tähän tutkimukseen poimittiin vain osa kasvilajeista, tuloksia on hankala vertailla muihin samaa aineistoa hyödyntäviin tutkimuksiin. Jonkinlaista osviittaa luokittelutulosten muuttumisesta lajimäärän kasvaessa voi saada Nilsbackin ja Zimmermanin (2006) artikkelista. He käyttivät analyyseissaan joko Oxfordin 17 lajin kuvakokoelman kaikkia lajeja tai toisena vaihtoehtona valitsivat kokoelmasta 10 lajia ja näistä 40 kuvaa, jotka olivat kuvakulmaltaan yhteneväisiä ja joissa kohde kattoi suurimman osan kuva-alasta. Kun valikoidusta kuva-aineistosta siirryttiin kaikki kuvat sisältävään kokoelmaan tunnistustarkkuus pieneni noin 17 prosenttiyksikön verran 81,3 %:iin. Tutkimuksen luokittelumenetelmänä oli lähimmän naapurin menetelmä, ja tunnistustarkkuus laskettiin viiden lähimmän naapurin painotettuna osumatarkkuutena. Seeland ym.:n (2017) vertailututkimuksessa mukana olleiden tutkimuksien paras luokittelutulos koko 17 lajin aineistolle oli 91,4 % (Xie ym. 2014).

Kotsiantis (2007) on koonnut katsausartikkeliinsa keskeisimpiä luokittelumenetelmiä ja niiden välisiä eroja. Bayes-verkkojen hyvänä puolena ja erona muihin menetelmiin on mahdollisuus hyödyntää prioritietoa esimerkiksi verkkorakenteita koskien. Toisin kuin neuroverkoista ja tukivektorikoneista, Bayes-verkosta on myös suoraan havaittavissa, miten luokitteluprosessi on rakentunut ja miten muuttujat vaikuttavat toisiinsa. Joissakin tapauksissa riip-

puvuudet verkon muuttujien välillä voidaan tulkita myös kausaalisuhteina ja hyödyntää tätä ominaisuutta tutkimuksissa (Charniak 1991; Tsamardinos, Brown ja Aliferis 2006). Neuroverkkoihin ja tukivektorikoneisiin verrattuna erityisesti naiivi Bayes-luokittelija on helppo mallintaa, sillä piirrevektorien jakaumien parametriestimaatit saadaan suoraan havaintoaineistosta. Myös havaintojen tarve on vähäinen (Kotsiantis 2007). Mutta kuten tässä tutkimuksessa on jo aiemmin käynyt ilmi, rakenteeltaan rajoittamattomissa, yleisissä Bayes-verkoissa rakenteen estimointi on NP-täydellinen ongelma (ks. luku 3.3.2), eikä mutkikkaammissa malleissa havaintojen määrä enää välttämättä riitä mallin luotettavaan estimointiin (ks. luku 4.5.4). Muihin menetelmiin verrattuna Bayes-verkkojen hyvänä puolena on myös se, että ne pystyvät hyvin käsittelemään puuttuvia havaintoja: piirrettä, josta havainto puuttuu, ei yksinkertaisesti käytetä luokan ennustamisessa. Myös mallin päivittäminen uusilla havainnoilla on helppoa (Kotsiantis 2007). Toisaalta generoiviin malleihin kuuluvana Bayes-verkkojen tarkkuus luokittelijoina ei ole välttämättä niin hyvä kuin tukivektorikoneen tai neuroverkon. Luokittelun kannalta redundantit tai muuten turhat piirrevektorit saattavat Bayes-verkoissa häiritä enemmän luokittelua kuin tukivektorikoneissa (Kotsiantis 2007). Toisaalta Bayes-verkkojen rakenteen estimointia voidaan hyödyntää valitsemaan sopivat piirrevektorit ja näin poistaa turhat muuttujat mallista (Tsamardinos, Brown ja Aliferis 2006).

Piirteinä tässä pro gradu -työssä käytettiin vain kukan väriä ja muotoa kuvaavia piirteitä. Tekstuurin hyödyntämistä pohdittiin, mutta se tuntui ongelmalliselta tämäntyyppisessä aineistossa, jossa kuvien terävyys- ja skaalavaihtelut ovat suuria ja iso osa kuvista sisältää runsaasti muun muassa pakkauksen tai skannauksen aiheuttamia artefakteja. Lisäksi, jos tarkastellaan kukan terälehtiä, usealla lajilla pääasiallisen tekstuurin muodostavat kukan keskustaa kohti kulkevat suonet, jotka muodostavat hyvin samanlaisen rakenteen lajista riippumatta. Nilsback ja Zisserman (2006) tekivät samansuuntaisen huomion tekstuurin luokittelukyvystä. Kymmenen lajin valikoidussa aineistossa muotopiirteet kykenivät parhaiten erottelemaan lajit toisistaan värin ja tekstuurin ollessa jotakuinkin yhtä hyviä. Sen sijaan kaikki kuvat sisältävässä aineistossa väri oli paras ja tekstuuri selvästi huonoin erottelija. Nilsback ja Zisserman (2006) arvioivatkin juuri kohteen skaalan suurien vaihteluiden ja selkeän havaittavan tekstuurin puuttumisen huonontaneen tekstuurin luokittelukykyä.

Kukan reunaosan avainpisteet jaettiin koon perusteella isoihin ja pieniin avainpisteisiin, joista edelleen määritettiin algoritmin avulla kukan terälehtiä potentiaalisesti vastaavat avainpisteet. Tarkoituksena oli pyrkiä kuvaamaan terälehtien kokoa ja sijaintia toisiinsa nähden. Pienien terälehtien luokan avainpisteiden kulmapiirteet erottelivat varsin hyvin lajeja toisistaan, mutta isojen terälehtien luokan avainpisteiden piirteet olivat ongelmallisia muun muassa puuttuvien havaintojen vuoksi. Saattaisikin riittää tutkia yhtäaikaa kaikenkokoisia reunaalueille sijoittuvia avainpisteitä ja niiden sijaintia toisiinsa nähden, koska algoritmi joka tapauksessa poistaisi suurimman osan keskemmälle sijoittuvista isommista avainpisteistä. Sen sijaan jos kuvat olisivat tämän tutkimuksen aineistoa parempilaatuisia ja kukan keskustasta löytyisi siten enemmän avainpisteitä, saattaisi olla hyödyllistä tarkastella erikseen niiden sijaintia ja laskea myös avainpisteiden deskriptoripiirteet sekä keskus- että reunaosalle. Keskustan ominaisuuksien parempi hyödyntäminen toki vaatisi, että kukat olisi kuvattu edestäpäin. Joka tapauksessa sekä kukan keskustan ominaisuuksia että terälehtien kokoa, muotoa ja sijaintia pitäisi pystyä kuvaamaan paremmin. Nilsback ja Zisserman (2007) pohtivat kukan muotomallia hyödyntäneessä, segmentointia käsitelleessä artikkelissaan, että muotomallin avulla voitaisiin pyrkiä erottelemaan myös yksittäisiä terälehtiä.

Toisaalta kukkien tapauksessa voidaan pohtia, mikä täsmälleen on se kokonaisuus, jota tulisi tarkastella. Esimerkiksi koiranputkilla ja monilla muilla sarjakukkaisilla kasveilla kukinto on kolmesta tasosta muodostuva kerrannaissarja: koko kukinto muodostuu erillisistä, yksittäisten pienien kukkien muodostamista kukkaryhmistä (esim. Nylén 1993). Yhtälailla kuin yksittäinen kukka, myös kukinnon yleismuoto ja kukkien sijoittuminen toisiinsa nähden kukinnossa voivat olla lajille tyypillisiä. Yksittäisen kukan muoto-ominaisuuksien ohella tulisikin tietyillä lajeilla tarkastella myös kukinnon muotoa. Toisaalta riippuu paljon aineistosta, kuinka yksityiskohtainen malli kukasta tai kukinnosta voidaan luoda. Esimerkiksi Bouchard ja Triggs (2005) päätyivät hierarkkisen osapohjaisen mallin rakentamisessa käyttämään vain yhtä koulutuskuvaa, jonka katsottiin parhaiten edustavan kohdetta.

Tutkielman aihe tuntui kokonaisuudessaan liian laajalta käytettävissä olevaan aikaan nähden. Työ sisälsi esimerkiksi melko paljon erilaisia menetelmiä, joihin perehtymiseen kului aikaa, vaikka osa niistä olikin ennalta tuttuja. Aikaa kului paljon esimerkiksi avainpisteiden etsimiseen ja erilaisiin kokeiluihin, miten avainpistedeskriptoreja tulisi muokata ja yhdistel-

lä. Raportissa mainittujen menetelmien lisäksi deskriptoreja kokeiltiin esimerkiksi klusteroida niin pehmeän klusteroinnin kuin tiheysperustaisella DBSCAN-menetelmälläkin. Selkeitä, deskriptoreja erottelevia klustereita ei kuitenkaan saatu. Materiaalipaljouden ja harharetkien vuoksi yksittäisiin osa-alueisiin ei ehtinyt perehtyä niin syvällisesti kuin olisi ollut tarpeen. Esimerkiksi koostemuuttujamalleja (ks. luku 4.7) rakennettaessa olisi ollut syytä perehtyä myös latenttien muuttujien malleihin (Bishop 1999). Myös puuttuvat havainnot olisivat periaatteessa vaatineet oman käsittelynsä estimointivaiheessa (Heckerman 1999). Yleensäkin Bayes-verkkojen erilaisiin muunnelmiin perehtyminen olisi vaatinut lisää aikaa. Yksi vaihtoehto olisikin ollut käyttää malleissa etukäteen laskettuja piirteitä. Näin jälkiviisaana erilaisten mallien kokeilu olisi kannattanut tehdä siten, että aluksi naiiveilla malleilla olisi tutkittu piirremuuttujien jakaumaluokkien sopivaa määrää luokittelutuloksiin ja sopivuuskriteerin arvoihin perustuen. Tämän jälkeen malliin olisi valittu sopivat piirteet rakenne-estimoinnin yhteydessä sekä tarkastelemalla piirteiden riippuvuuksia luokkamuuttujasta. Vasta lopuksi olisi toteutettu luokittelu hyödyntäen FAN-malleja.

Tutkielman huono puoli oli samalla myös sen hyvä puoli. Kun mukana olivat sekä piirteiden muodostaminen että mallien rakentaminen, tutkimus kattoi koko konenäkötutkimukseen kuuluvan syklin ja antoi kattavan kokonaiskuvan konenäkötutkimuksen teosta. Johdannossa tutkimuksen mainitaan olevan design science -tyyppistä tutkimusta. Sitä kuvaillaan johdantoluvussa seuraavasti:

> Design science -tutkimukseen sisältyy olennaisena osana kehittämis- ja evaluointisykli, joka luonteeltaan iteratiivisena auttaa omalta osaltaan tutkijaa paremmin ymmärtämään tarkasteltavaa ongelmaa (Hevner ym. 2004).

Tältä osin tutkimuksen tavoitteen voidaan arvioida täyttyneen hyvin.

Lähteet

Abdel-Hakim, Alaa E., ja Aly A. Farag. 2006. “CSIFT: A SIFT descriptor with color invariant characteristics”. Teoksessa *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* 2:1978–1983. IEEE. doi:10.1109/CVPR.2006.95.

Ankan, Ankur, ja Abinash Panda. 2015. *Mastering Probabilistic Graphical Models Using Python: Master probabilistic graphical models by learning through real-world problems and illustrative code examples in Python.* Packt Publishing.

Arlot, Sylvain, ja Alain Celisse. 2010. “A survey of cross-validation procedures for model selection”. *Statistics Surveys* 4:40–79. doi:10.1214/09-SS054.

Bay, Herbert, Andreas Ess, Tinne Tuytelaars ja Luc Van Gool. 2008. “Speeded-up robust features (SURF)”. *Computer Vision and Image Understanding* 110:346–359. doi:10.1016/j.cviu.2007.09.014.

Bishop, Christopher M. 1999. “Latent variable models”. Teoksessa Jordan 1999, 371–403.

Boas, Mary L. 1983. *Mathematical Methods in the Physical Sciences.* John Wiley & Sons.

Bouchard, Guillaume, ja Bill Triggs. 2005. “Hierarchical part-based visual object categorization”. Teoksessa *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* 1:701–715. IEEE. doi:10.1109/CVPR.2005.174.

Boureau, Y-Lan, Francis Bach, Yann LeCun ja Jean Ponce. 2010. “Learning mid-level features for recognition”. Teoksessa *Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* 2559–2566. IEEE. doi:10.1109/CVPR.2010.5539963.

Boykov, Yuri Y., ja Marie-Pierre Jolly. 2001. “Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images”. Teoksessa *Proceedings Eighth IEEE International Conference on Computer Vision,* 1:105–112. IEEE. doi:10.1109/ICCV.2001.937505.

Carneiro, Gustavo, ja David Lowe. 2006. “Sparse flexible models of local features”. Teoksessa *Computer Vision – ECCV 2006. ECCV 2006. Lecture Notes in Computer Science,* 3953:29–43. Springer, Berlin, Heidelberg. doi:10.1007/11744078_3.

Charniak, Eugene. 1991. “Bayesian networks without tears”. *AI Magazine* 12 (4): 50–63. doi:10.1609/aimag.v12i4.918.

Cheng, Jie, ja Russell Greiner. 1999. “Comparing Bayesian network classifiers”. Teoksessa *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI1999),* 101–108. Morgan Kaufmann Publishers Inc. https://arxiv.org/pdf/1301.6684.

Chow, C., ja C. Liu. 1968. “Approximating discrete probability distributions with dependence trees”. Teoksessa *IEEE Transactions on Information Theory,* 14:462–467. IEEE. doi:10.1109/TIT.1968.1054142.

Cooper, Gregory F., ja Edward Herskovits. 1992. “A Bayesian method for the induction of probabalistic networks from data”. *Machine Learning* 9 (4): 309–347. doi:10.1023/A:1022649401552.

Cowell, Robert. 1999. “Introduction to inference for Bayesian networks”. Teoksessa Jordan 1999, 9–26.

Csurka, Gabriella, Christopher R. Dance, Lixin Fan, Jutta Willamowski ja Cédric Bray. 2004. “Visual categorization with bags of keypoints”. Teoksessa *Workshop on Statistical Learning in Computer Vision, ECCV,* 1–22. Viitattu 6. lokakuuta 2018. https://people.eecs.berkeley.edu/~efros/courses/AP06/Papers/csurka-eccv-04.pdf.

Dechter, R. 1999. “Bucket elimination: A unifying framework for probabilistic inference”. Teoksessa Jordan 1999, 75–104.

Domingos, Pedro, ja Michael Pazzani. 1996. “Beyond independence: Conditions for the optimality of the simple Bayesian classifier”. Teoksessa *Proceedings of the Thirteenth International Conference on International Conference on Machine Learning,* 105–112. Morgan Kaufmann.

Donchenko, Volodymyr, ja Andrew Golik. 2013. “Matrix feature vectors and Hu moments in gesture recognition”. *International Journal "Information Technologies & Knowledge"* 7 (4): 380–391.

Fergus, Rob, Pietro Perona ja Andrew Zisserman. 2005. “A sparse object category model for efficient learning and exhaustive recognition”. Teoksessa *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* 1:380–387. IEEE. doi:10.1109/CVPR.2005.47.

Flusser, Jan. 2000. “On the independence of rotation moment invariants”. *Pattern Recognition* 33 (9): 1045–1410. doi:10.1016/S0031-3203(99)00127-2.

Friedman, Nir, Dan Geiger ja Moises Goldszmidt. 1997. “Bayesian network classifiers”. *Machine Learning* 29 (2–3): 131–163. doi:10.1023/A:1007465528199.

Ganesan, P., V. Rajini ja R. Immanuvel Rajkumar. 2010. “Segmentation and edge detection of color images using CIELAB color space and edge detectors”. Teoksessa *Emerging Trends in Robotics and Communication Technologies (INTERACT-2010),* 393–397. IEEE. doi:10.1109/INTERACT.2010.5706186.

Hamine, Vikas, ja Paul Helman. 2005. “Learning optimal augmented Bayes networks”. Viitattu 21. toukokuuta 2018. https://arxiv.org/pdf/cs/0509055.pdf.

Hansen, Dennis M., Timotheüs Van der Niet ja Steven D. Johnson. 2012. “Floral signposts: Testing the significance of visual ’nectar guides’ for pollinator behaviour and plant fitness”. *Proceedings of the Royal Society B: Biological sciences* 279 (1729). doi:10.1098/rspb.2011.1349.

Heckerman, David. 1999. “A tutorial on learning with Bayesian networks”. Teoksessa Jordan 1999, 301–354.

Hevner, Alan, Salvatore March, Jinsoo Park ja Sudha Ram. 2004. “Design science in information systems research”. *MIS Quarterly* 28 (1): 75–105.

Hietanen, Antti, Jukka Lankinen, Joni-Kristian Kämäräinen, Anders Glent Buch ja Norbert Krüger. 2016. “A comparison of feature detectors and descriptors for object class matching”. *Neurocomputing* 184:3–12.

Hu, Ming-Kuei. 1962. “Visual pattern recognition by moment invariants”. *IRE Transactions on Information Theory* 8 (2): 179–187. doi:10.1109/TIT.1962.1057692.

Jackson, Donald A. 1993. “Stopping rules in principal components analysis: A comparison of heuristical and statistical approaches”. *Ecology* 74 (8). doi:10.2307/1939574.

Jayech, Khlifia, ja Mohamed Ali Mahjoub. 2010. “New approach using Bayesian network to improve content based image classification systems”. *International Journal of Computer Science Issues* 7 (6): 53–62.

Jordan, Michael. 1999. *Learning in Graphical Models.* MIT Press.

Keogh, Eamonn J., ja Michael J. Pazzani. 1999. “Learning augmented Bayesian classifiers: A comparison of distribution-based and classification-based approaches”. Viitattu 9. elokuuta 2018. https://www.ics.uci.edu/~pazzani/Publications/EamonnAIStats.pdf.

Kotsiantis, Sotiris. 2007. “Supervised machine learning: A review of classification techniques”. *Informatica (Ljubljana)* 31:249–268.

Kruskal, Joseph B. 1956. “On the shortest spanning subtree of a graph and the traveling salesman problem”. *Proceedings of the American Mathematical Society* 7:48–50. doi:10.1090/S0002-9939-1956-0078686-7.

Latham, Peter, ja Yasser Roudi. 2009. “Mutual information”. *Scholarpedia* 4 (1): 1658. doi:10.4249/scholarpedia.1658.

Liao, Simon Xinmeng. 1993. “Image analysis by moments”. Tohtorinväitöskirja, University of Manitoba. https://mspace.lib.umanitoba.ca/bitstream/handle/1993/18179/Liao_Image_analysis.pdf.

Loesdau, Martin, Sébastien Chabrier ja Alban Gabillon. 2014. “Hue and saturation in the RGB color space”. Teoksessa *Lecture Notes in Computer Science,* 8509:203–212. Springer. doi:10.1007/978-3-319-07998-1_23.

Lowe, David. 1999. “Object recognition from local scale-invariant features”. Teoksessa *Proceedings of the Seventh IEEE International Conference on Computer Vision,* nide 2. IEEE. doi:10.1109/ICCV.1999.790410.

Lowe, David. 2004. “Distinctive image features from scale-invariant keypoints”. *International Journal of Computer Vision* 60:91–110. doi:10.1023/B:VISI.0000029664.99615.94.

Lucas, Peter J.F. 2002. “Restricted Bayesian network structure learning”. Teoksessa *PGM 2002: Proceedings of the first European workshop on Probabilistic Graphical Models,* 117–126. UCLM. http://hdl.handle.net/2066/176093.

Melander, Markus, Petri Hienonen, Tuula Karhunen ja Laura Soosalu. 2016. *Konenäkö ja automatisoitu tiedon tuottaminen viheralueista: Inventointipilotti 2016,* Liikenneviraston tutkimuksia ja selvityksiä 55. https://julkaisut.liikennevirasto.fi/pdf8/lts_2016-55_konenako_automatisoitu_web.pdf.

Myllymäki, Petri, ja Henry Tirri. 1998. *Bayes-verkkojen mahdollisuudet,* Teknologiakatsaus 58. Tekes. https://www.cs.helsinki.fi/u/myllymak/bvmahd.pdf.

Ng, Andrew Y., ja Michael I. Jordan. 2002. “On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes”. Teoksessa *Proceedings of the Conference on Advances in Neural Information Processing Systems 14,* 841–848. MIT Press.

Nilsback, Maria-Elena, ja Andrew Zisserman. 2006. “A visual vocabulary for flower classification”. Teoksessa *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* 2:1447–1454. IEEE. doi:10.1109/CVPR.2006.42.

———. 2007. “Delving into the whorl of flower segmentation”. Teoksessa *Proceedings of the British Machine Vision Conference,* 1:570–579. BMVA Press. doi:10.5244/C.21.54.

———. 2008. “Automated flower classification over a large number of classes”. Teoksessa *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing.* IEEE. doi:10.1109/ICVGIP.2008.47.

Nylén, Bo. 1993. *Suomen ja Pohjolan kasvit.* WSOY.

Ranta, Esa, Hannu Rita ja Jari Kouki. 1997. *Biometria — Tilastotiedettä ekologeille.* Yliopistopaino.

Rubinstein, Y. Dan, ja Trevor Hastie. 1997. “Discriminative vs informative learning”. Teoksessa *Proceedings of the International Conference on Knowledge Discovery and Data Mining,* 49–53. AAAI Press. doi:10.1109/CVPR.2005.174.

Seeland, M., M. Rzanny, N. Alaqraa, J. Wäldchen ja P. Mäder. 2017. “Plant species classification using flower images — A comparative study of local feature representations”. *PLoS ONE* 12 (2): 1–29. doi:10.1371/journal.pone.0170629.

Szeliski, Richard. 2011. *Computer vision: Algorithms and applications.* Springer. Viitattu 16. syyskuuta 2017. http://szeliski.org/Book/.

Tibshirani, Robert, Guenther Walther ja Trevor Hastie. 2002. “Estimating the number of clusters in a data set via the gap statistic”. *Journal of the Royal Statistical Society: Series B* 63 (2): 411–423. doi:10.1111/1467-9868.00293.

Tsamardinos, Ioannis, Laura E. Brown ja Constantin F. Aliferis. 2006. “The max-min hill-climbing Bayesian network structure learning algorithm”. *Machine Learning* 65:31–78. doi:10.1007/s10994-006-6889-7.

Tukey, John W. 1977. *Exploratory Data Analysis.* Reading, Mass.

Vedaldi, A., ja B. Fulkerson. 2008. *VLFeat: An open and portable library of computer vision algorithms.* http://www.vlfeat.org/. Viitattu 16. helmikuuta 2018.

Wäldchen, Jana, ja Patrick Mäder. 2017. “Plant species identification using computer vision techniques: A systematic literature review”. *Archives of Computational Methods in Engineering*. doi:10.1007/s11831-016-9206-z.

Xie, Lingxi, Qi Tian, Meng Wang ja Bo Zhang. 2014. “Spatial pooling of heterogeneous features for image classification”. *IEEE Transactions on Image Processing* 23 (5): 1994–2008. doi:10.1109/TIP.2014.2310117.

Xue, Jiang-Hao. 2008. “Aspects of generative and discriminative classifiers”. Tohtorinväitöskirja, University of Glasgow. http://theses.gla.ac.uk/id/eprint/272.

Yousefi, Jamileh. 2011. “Image binarization using otsu thresholding algorithm”. Viitattu 14. lokakuuta 2017. doi:10.13140/RG.2.1.4758.9284.

Zaki, Mohammed J., ja Wagner Meira. 2014. *Data Mining and Analysis: Fundamental Concepts and Algorithms.* Cambridge University Press. Viitattu 9. lokakuuta 2017. http://www.dataminingbook.info/pmwiki.php/Main/HomePage.

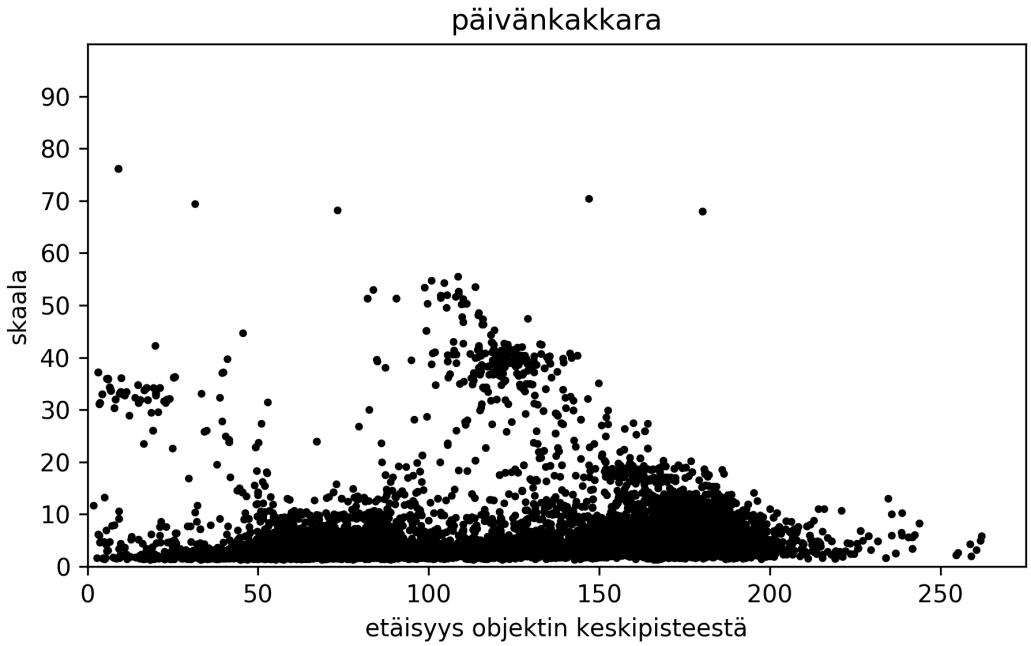
Zhang, Harry. 2004. "The optimality of naive Bayes". Teoksessa *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference,* 562–567.

Zhang, Xin, Yee-Hong Yang, Zhiguang Han, Hui Wang ja Chao Gao. 2013. "Object class detection: A survey". *ACM Computing Surveys (CSUR)* 46 (1). doi:10.1145/2522968.2522978.

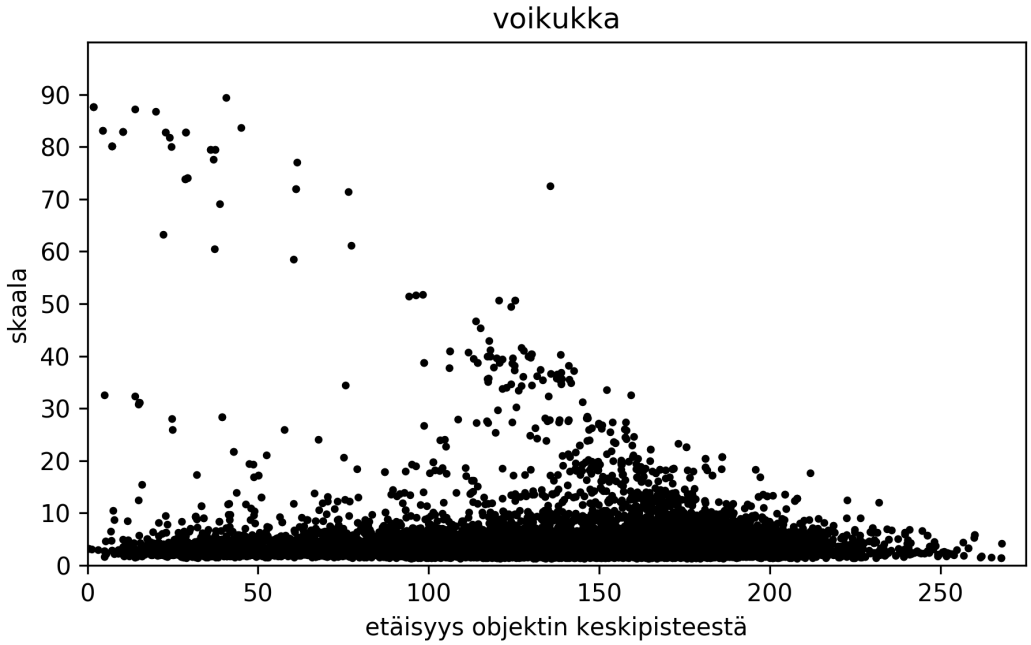
Zhu, Song-Chun, ja David Mumford. 2006. "A stochastic grammar of images". *Foundations and Trends®in Computer Graphics and Vision* 2 (4): 259–362. doi:10.1561/0600000018.

Liitteet

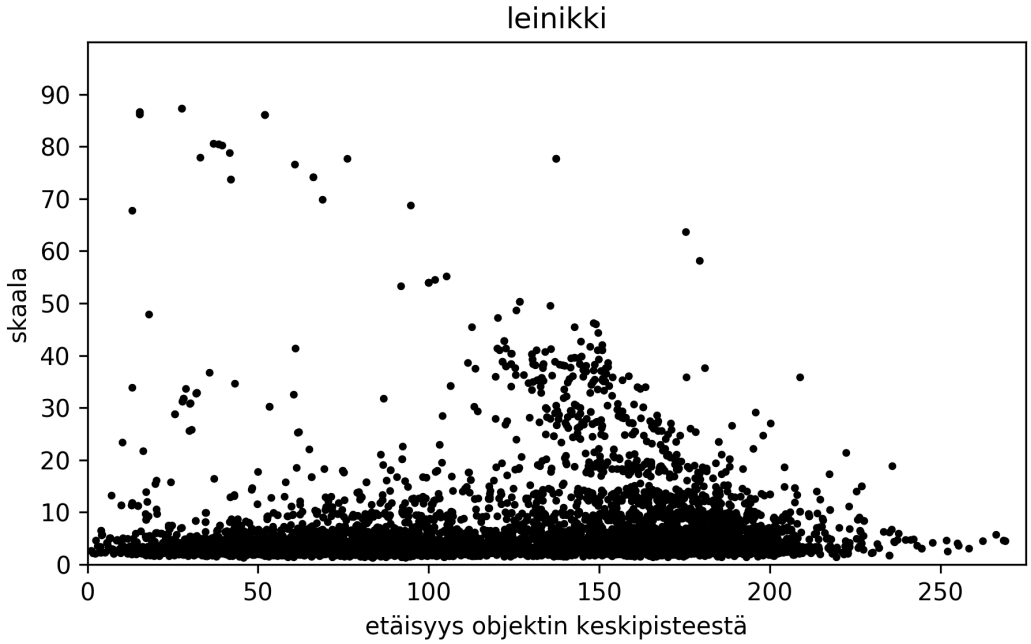
A Avainpisteiden skaalat ja etäisyydet lajeittain



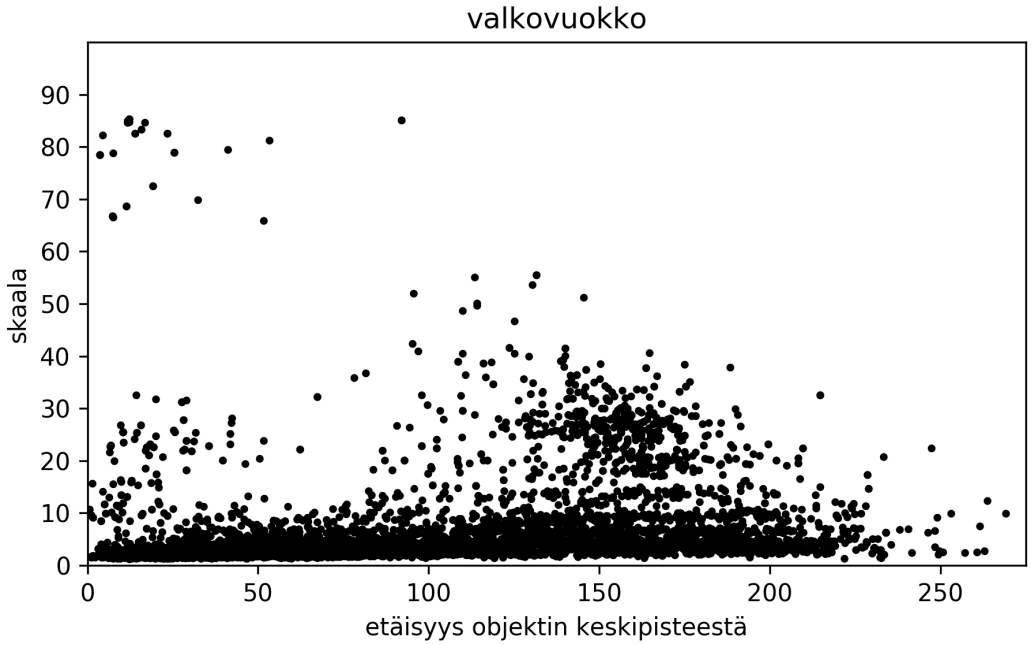
Kuvio 29: Päivänkakkaroista löydetyt avainpisteet skaalan ja kohteen keskipisteestä mitatun etäisyyden suhteen kuvattuna.



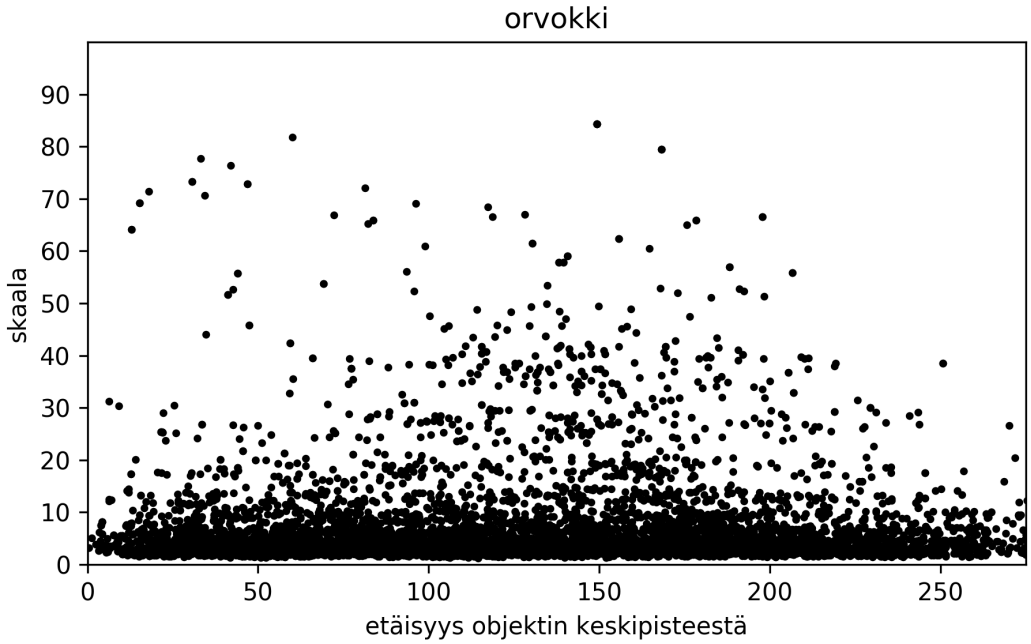
Kuvio 30: Voikukista löydetyt avainpisteet skaalan ja kohteen keskipisteestä mitatun etäisyyden suhteen kuvattuna.



Kuvio 31: Leinikeistä löydetyt avainpisteet skaalan ja kohteen keskipisteestä mitatun etäisyyden suhteen kuvattuna.



Kuvio 32: Valkovuokoista löydetyt avainpisteet skaalan ja kohteen keskipisteestä mitatun etäisyyden suhteen kuvattuna.



Kuvio 33: Orvokeista löydetyt avainpisteet skaalan ja kohteen keskipisteestä mitatun etäisyyden suhteen kuvattuna.

B FAN-mallien sekaannusmatriisit

Taulukko 18: Metsärakenteella täydennetyn naiivin Bayes-mallin sekaannusmatriisi (%), kun malliin valittujen kaarien painon raja-arvona on piirremuuttujien ehdollisen yhteisen informaation keskiarvo.

	Päivänkakkara	Voikukka	Leinikki	Valkovuokko	Orvokki
Päivänkakkara	83,8	1,3	0,0	13,8	1,3
Voikukka	2,5	91,3	6,3	0,0	0,0
Leinikki	2,5	7,5	86,3	3,8	0,0
Valkovuokko	10,0	0,0	1,3	86,3	2,5
Orvokki	7,5	1,3	5,0	7,5	78,8

Taulukko 19: Metsärakenteella täydennetyn naiivin Bayes-mallin sekaannusmatriisi (%), kun malliin valittujen kaarien painon raja-arvona on piirremuuttujien ehdollisen mutuaalisen informaation 90. persentiili.

	Päivänkakkara	Voikukka	Leinikki	Valkovuokko	Orvokki
Päivänkakkara	86,3	0,0	0,0	11,3	2,5
Voikukka	1,3	95,0	3,8	0,0	0,0
Leinikki	1,3	6,3	87,5	3,8	1,3
Valkovuokko	5,0	0,0	0,0	92,5	2,5
Orvokki	3,8	1,3	5,0	8,8	81,3

Taulukko 20: Metsärakenteella täydennetyn naiivin Bayes -mallin sekaannusmatriisi (%), kun malliin valittujen kaarien painon raja-arvona on piirremuuttujien ehdollisen mutuaalisen informaation 95. persentiili.

	Päivänkakkara	Voikukka	Leinikki	Valkovuokko	Orvokki
Päivänkakkara	86,3	0,0	2,5	8,8	2,5
Voikukka	0,0	95,0	3,8	0,0	1,3
Leinikki	1,3	5,0	86,3	5,0	2,5
Valkovuokko	6,3	0,0	0,0	90,0	3,8
Orvokki	0,0	0,0	3,8	3,8	92,5

Taulukko 21: Metsärakenteella täydennetyn naiivin Bayes-mallin sekaannusmatriisi (%), kun malliin valittujen kaarien painon raja-arvona on piirremuuttujien ehdollisen mutuaalisen informaation 99. persentiili.

	Päivänkakkara	Voikukka	Leinikki	Valkovuokko	Orvokki
Päivänkakkara	87,5	1,3	1,3	7,5	2,5
Voikukka	0,0	97,5	2,5	0,0	0,0
Leinikki	3,8	7,5	88,8	0,0	0,0
Valkovuokko	5,0	0,0	1,3	92,5	1,2
Orvokki	0,0	0,0	1,3	3,8	95,0

C Rakenteeltaan rajoittamattomien mallien sekaannusmatriisit

Malleissa mukana olevat piirremuuttujat ja rakenteet on kuvattu taulukossa 10.

Taulukko 22: Rajoittamattoman K2-naive-mallin sekaannusmatriisi (%).

	Päivänkakkara	Voikukka	Leinikki	Valkovuokko	Orvokki
Päivänkakkara	88,8	0,0	1,3	6,3	3,8
Voikukka	0,0	97,5	1,3	0,0	1,3
Leinikki	1,3	3,8	95,0	0,0	0,0
Valkovuokko	3,8	0,0	0,0	95,0	1,3
Orvokki	0,0	0,0	0,0	2,5	97,5

Taulukko 23: Rajoittamattoman BIC-naive-mallin sekaannusmatriisi (%).

	Päivänkakkara	Voikukka	Leinikki	Valkovuokko	Orvokki
Päivänkakkara	68,8	2,5	0,0	26,3	2,5
Voikukka	0,0	76,3	21,3	0,0	2,5
Leinikki	0,0	30,0	68,8	0,0	1,3
Valkovuokko	26,3	0,0	0,0	67,5	6,3
Orvokki	5,0	5,0	0,0	5,0	85,0

Taulukko 24: Rajoittamattoman K2-empty-mallin sekaannusmatriisi (%).

	Päivänkakkara	Voikukka	Leinikki	Valkovuokko	Orvokki
Päivänkakkara	90,0	0,0	0,0	5,0	5,0
Voikukka	0,0	96,3	3,8	0,0	0,0
Leinikki	1,3	6,3	92,5	0,0	0,0
Valkovuokko	8,8	0,0	0,0	91,3	0,0
Orvokki	0,0	1,3	1,3	5,0	92,5

Taulukko 25: Rajoittamattoman K2b-empty-mallin sekaannusmatriisi (%).

	Päivänkakkara	Voikukka	Leinikki	Valkovuokko	Orvokki
Päivänkakkara	90,0	0,0	1,3	5,0	3,8
Voikukka	0,0	100,0	0,0	0,0	0,0
Leinikki	1,3	7,5	91,3	0,0	0,0
Valkovuokko	5,0	0,0	0,0	95,0	0,0
Orvokki	0,0	0,0	0,0	2,5	97,5

Taulukko 26: Naive-K2b-mallin sekaannusmatriisi (%).

	Päivänkakkara	Voikukka	Leinikki	Valkovuokko	Orvokki
Päivänkakkara	88,8	0,0	1,3	6,3	3,8
Voikukka	0,0	98,8	0,0	0,0	1,3
Leinikki	1,3	5,0	93,8	0,0	0,0
Valkovuokko	5,0	0,0	0,0	95,0	0,0
Orvokki	0,0	0,0	0,0	1,3	98,8

Taulukko 27: Rajoittamattoman K2b-2C5-mallin sekaannusmatriisi (%).

	Päivänkakkara	Voikukka	Leinikki	Valkovuokko	Orvokki
Päivänkakkara	86,3	0,0	3,8	6,3	3,8
Voikukka	0,0	95,0	1,3	0,0	3,8
Leinikki	1,3	2,5	91,3	0,0	5,0
Valkovuokko	3,8	0,0	3,8	92,5	0,0
Orvokki	0,0	0,0	5,0	5,0	90,0

Taulukko 28: Rajoittamattoman K2b-1C5-mallin sekaannusmatriisi (%).

	Päivänkakkara	Voikukka	Leinikki	Valkovuokko	Orvokki
Päivänkakkara	90,0	0,0	0,0	6,3	3,8
Voikukka	0,0	98,8	0,0	0,0	1,3
Leinikki	0,0	2,5	95,0	0,0	2,5
Valkovuokko	5,0	0,0	0,0	95,0	0,0
Orvokki	2,5	0,0	0,0	1,3	96,3

Taulukko 29: Naive-K2b-C5-mallin sekaannusmatriisi (%).

	Päivänkakkara	Voikukka	Leinikki	Valkovuokko	Orvokki
Päivänkakkara	88,8	0,0	0,0	7,5	3,8
Voikukka	0,0	97,5	1,3	0,0	1,3
Leinikki	1,3	7,5	91,3	0,0	0,0
Valkovuokko	3,8	0,0	0,0	95,0	1,3
Orvokki	0,0	0,0	1,3	1,3	97,5

Taulukko 30: Rajoittamattoman BICb-2C5-mallin sekaannusmatriisi (%).

	Päivänkakkara	Voikukka	Leinikki	Valkovuokko	Orvokki
Päivänkakkara	90,0	0,0	0,0	6,3	3,8
Voikukka	0,0	98,8	0,0	0,0	1,3
Leinikki	0,0	1,3	97,5	0,0	1,3
Valkovuokko	7,5	0,0	0,0	92,5	0,0
Orvokki	1,3	0,0	0,0	3,8	95,0

Taulukko 31: Naive-BICb-C5-mallin sekaannusmatriisi (%).

	Päivänkakkara	Voikukka	Leinikki	Valkovuokko	Orvokki
Päivänkakkara	90,0	0,0	0,0	6,3	3,8
Voikukka	0,0	96,3	2,5	0,0	1,3
Leinikki	1,3	3,8	95,0	0,0	0,0
Valkovuokko	6,3	0,0	0,0	92,5	1,3
Orvokki	0,0	0,0	1,3	3,8	95,0

D Koostemuuttujia sisältävien mallien sekaannusmatriisit

Taulukko 32: Koostemuuttujia sisältävän Aw-mallin sekaannusmatriisi (%). Mukana ovat kaikki piirremuuttujat ja koostemuuttuja on laskettu painotettuna summana.

	Päivänkakkara	Voikukka	Leinikki	Valkovuokko	Orvokki
Päivänkakkara	88,8	1,3	1,3	5,0	3,8
Voikukka	0,0	98,8	0,0	0,0	1,3
Leinikki	1,3	7,5	91,3	0,0	0,0
Valkovuokko	5,0	0,0	0,0	93,8	1,3
Orvokki	0,0	0,0	1,3	2,5	96,3

Taulukko 33: Koostemuuttujia sisältävän A-mallin sekaannusmatriisi (%). Mukana ovat kaikki piirremuuttujat. Koostemuuttujan piirteissä ei ole painotusta.

	Päivänkakkara	Voikukka	Leinikki	Valkovuokko	Orvokki
Päivänkakkara	90,0	1,3	3,8	3,8	1,3
Voikukka	0,0	93,8	2,5	0,0	3,8
Leinikki	2,5	10,0	81,3	1,3	5,0
Valkovuokko	5,0	0,0	3,8	91,3	0,0
Orvokki	1,3	1,3	7,5	1,3	88,8

Taulukko 34: Koostemuuttujia sisältävän B1w-C20-mallin sekaannusmatriisi (%). Piirremuuttujista on poistettu ison terälehtiluokan piirteet, deskriptorit d6 ja d7 sekä keskiosan pienien avainpisteiden momenttipiirre. Koostemuuttujat on laskettu painotettuna summana.

	Päivänkakkara	Voikukka	Leinikki	Valkovuokko	Orvokki
Päivänkakkara	91,3	1,3	1,3	2,5	3,8
Voikukka	0,0	97,5	2,5	0,0	0,0
Leinikki	1,3	6,3	92,5	0,0	0,0
Valkovuokko	3,8	0,0	0,0	96,3	0,0
Orvokki	0,0	0,0	1,3	1,3	97,5

Taulukko 35: Koostemuuttujia sisältävän B1-C20-mallin sekaannusmatriisi (%). Piirremuuttujista on poistettu ison terälehtiluokan piirteet, deskriptorit d6 ja d7 sekä keskiosan pienien avainpisteiden momenttipiirre. Koostemuuttujan piirteissä ei ole painotusta.

	Päivänkakkara	Voikukka	Leinikki	Valkovuokko	Orvokki
Päivänkakkara	87,5	3,8	2,5	2,5	3,8
Voikukka	0,0	97,5	2,5	0,0	0,0
Leinikki	6,3	7,5	80,0	1,3	5,0
Valkovuokko	3,8	0,0	6,3	88,8	1,3
Orvokki	2,5	1,25	8,8	5,0	82,5

Taulukko 36: Koostemuuttujia sisältävän B2w-C5-mallin sekaannusmatriisi (%). Piirremuuttujat ovat 5-luokkaisia ja niistä on poistettu ison terälehtiluokan piirteet. Koostemuuttujat on laskettu painotettuna summana.

	Päivänkakkara	Voikukka	Leinikki	Valkovuokko	Orvokki
Päivänkakkara	92,5	0,0	0,0	2,5	5,0
Voikukka	0,0	97,5	1,3	0,0	1,3
Leinikki	1,3	2,5	95,0	0,0	1,3
Valkovuokko	6,3	0,0	1,3	92,5	0,0
Orvokki	1,3	0,0	2,5	3,8	92,5

Taulukko 37: Koostemuuttujia sisältävän B1w-C5-mallin sekaannusmatriisi (%). Piirremuuttujat ovat 5-luokkaisia ja niistä on poistettu ison terälehtiluokan piirteet, deskriptorit d6 ja d7 sekä keskiosan pienien avainpisteiden momenttipiirre. Koostemuuttujat on laskettu painotettuna summana.

	Päivänkakkara	Voikukka	Leinikki	Valkovuokko	Orvokki
Päivänkakkara	93,8	0,0	0,0	1,3	5,0
Voikukka	0,0	97,5	1,3	0,0	1,3
Leinikki	2,5	0,0	97,5	0,0	0,0
Valkovuokko	5,0	0,0	1,3	93,8	0,0
Orvokki	1,3	0,0	2,5	3,8	92,5