Bayes-verkkojen mahdollisuudet

Petri Myllymäki   Henry Tirri

TEKES
Julkaisu xx/98
Helsinki 1998

Esipuhe

Alkusanat

Oppivien ja älykkäiden järjestelmien tutkimus on viime aikoina organisoitunut kokonaisuudeksi, josta käytetään myös nimiä *laskennallinen älykkyys* (computational intelligence) ja *soft computing.* Ajatuksena on, että vaikeissa päättelyongelmissa korvataan eksplisiittisten mallien ja ihmistietämyksen puute laskennallisilla (numeerisilla) malleilla, jotka sovitetaan tutkittavaan kohteeseen suurelta osin mittaustietoja hyväksikäyttäen. Tämä tietysti vain silloin, kun paremman puutteessa on pakko turvautua dataohjattuihin empiirisiin malleihin. Asiantuntijalla on kyllä näkemys kohteensa (esimerkiksi prosessi tai laite) toiminnasta, mutta tarkkojen ja ennustusvoimaisten mallien aikaansaamiseksi voi olla pakko täyttää puuttuvia osia soft computing-malleilla. Lopputulos on silloin monista osista koostuva *hybridimalli.*

Soft computing-alan valtavirrat ovat neuroverkot ja sumeat järjestelmät. Niillä on eniten sovelluksia niin Suomessa kuin muuallakin, alat ovat jo kypsyneet valmiiden ohjelmistopakettien ja hyvätasoisten oppikirjojen tasolle, ja runsaasti käyttökokemusta on olemassa neuroverkkojen tai sumeiden sääntöjen — tai molempien — soveltamisesta laajaan kirjoon käytännön ongelmia. Näitä kahta alaa sovelluksineen on selostettu Tekesin aikaisemmissa raporteissa "Sumean logiikan mahdollisuudet" (1993) ja "Neurolaskennan mahdollisuudet" (1994). Laskennalliseen älykkyyteen kuuluu kuitenkin muitakin metodiikkoja. Tärkeimmät ovat *evoluutiolaskenta*, etenkin geneettiset optimointialgoritmit, sekä *Bayes-verkot*, jota nyt käsillä oleva raportti käsittelee.

Voi tietenkin kysyä, onko yleensä mielekästä puhua tieteenalasta nimeltä laskennallinen älykkyys tai soft computing, joka määritellään tiettynä suljettuna menetelmäjoukkona. Varmaa ainakin on, että ala on voimakkaassa muutostilassa ja uusia teknologioita ja menetelmiä epäilemättä syntyy lähitulevaisuudessa, joten kuva on koko ajan muutostilassa. Eri menetelmien lähtökohdat ja sovellusmahdollisuudet ovat myös osittain erilaisia. Mielestäni — ja monen muunkin alan tutkijan mielestä — on järkevää niputtaa nämä uudet alat omaksi metodologiakseen tai tieteenalakseen, koska niillä on kuitenkin joukko yhteisiä piirteitä, jotka erottavat ne perinteisestä komputoinnista. Ensinnäkin niiden välillä on yhteyksiä: esimerkiksi neuroverkkoja käytetään virittämään sumeita järjestelmiä, geneettisillä algoritmeilla voi tehokkaasti optimoida neuroverkkojen rakennetta, ja Bayes-verkot antavat matemaattisesti perustellun ja siksi voimakkaan mallin samoihin päättelyongelmiin, joita ratkotaan neuroverkoilla tai sumeilla järjestelmillä. Toiseksi, ja ehkä vieläkin tärkeämpänä syynä, on menetelmien perusidea ja lähtökohta — enemmän tai vähemmän biologisista tai kognitiivisista prosesseista johdetaan toimintamalleja, jotka yksinkertaistettuina ja muunneltuina sovitetaan oppiviksi ja älykkäiksi ratkaisuiksi käytännön ongelmiin. Tärkeää on nimen omaan laskennallisuuden korostaminen: oppivat ja älykkäät järjestelmät, ainakin realistisissa isoissa ongelmissa, toimivat tehokkaidenkin tietokoneiden äärirajoilla, eikä niitä olisi voinut ajatellakaan käytännön menetelminä vielä kymmenenkään vuotta sitten. Tietokoneiden tehon edelleen kasvaessa tulemme varmasti näkemään uusia laskennallisesti älykkäitä menetelmiä, jotka aina käyttävät kaiken saatavilla olevan laskentatehon yhä parempien ja parempien päättelymallien rakentamiseksi.

Mitä soveltaja tästä alojen niputtamisesta sitten hyötyy? Suoranaisesti ei paljoakaan. Kuitenkin ne valmiit menetelmät, joita hän ongelmiinsa kokeilee tai käyttää, ovat syntyneet tutkimuksen tuloksena, jossa erityyppisten oppivien ja älykkäiden järjestelmien vuorovaikutus on antanut uusia ideoita ja johtanut parempiin tuloksiin.

Bayes-verkot ovat päättelyjärjestelmiä, jotka nojautuvat vanhaan ja koeteltuun todennäköisyyslaskentaan, etenkin ns. bayesiläiseen päättelyyn. Lyhyesti sanottuna tämä tarkoittaa, että kaikille mallituksen elementeille — itse malliarkkitehtuurille, sovitettaville parametreille ja sovitukseen käytettävälle datalle — oletetaan todennäköisyysjakaumat, ja valitsemme sen mallin ja ne parametrit, jotka ovat kaikkein todennäköisimmät ottaen huomioon saamamme mittausdatan. Ajatus on elegantti ja sallii asiantuntijan tietämyksen käyttämisen, koska hän voi kokemuksensa perusteella valita erilaisten mallien ja parametrien etukäteistodennäköisyydet (priorit). Vaikeutena toisaalta on, että todennäköisyysformalismi eleganttisuudessaan voi olla liiankin yleinen — suurta joukkoa tarvittavia todennäköisyysjakaumia on hankala estimoida ja niitä yhdistäviä hyvin moniulotteisia integraaleja on vaikea laskea. Tällöin joudutaan voimakkaisiin yksinkertaistuksiin, jotka murentavat mallien tyylikästä formalismia. Kuitenkin yksinkertaistetutkin mallit ovat monissa käytännön ongelmissa, esimerkiksi luokittelussa, osoittautuneet hyvin tehokkaiksi ja toimiviksi.

Geneettiset algoritmit ovat optimointimenetelmiä. Niissä pyritään löytämään jonkin annetun kohdefunktion optimaalinen minimi tai maksimi. Funktion ei tarvitse olla annettuna matemaattisessa muodossa, vaan pikemminkin menetelmä puree parhaiten ongelmiin joissa kohdefunktion arvot joudutaan laskemaan tai simuloimaan ehkä kalliistikin tietokoneohjelmilla. Kohdefunktio voi kuvata vaikkapa hissin ohjausjärjestelmää tai jonkin suunniteltavan koneenosan muodon hyvyyttä. Erona useimpiin muihin optimointimenetelmiin geneettinen algoritmi käyttää joukkoa — sukupolvea — rinnakkaisia ratkaisuyritteitä, joista tietyillä satunnaisilla menetelmillä yhdistellään vielä parempi sukupolvi uusia ratkaisuyritteitä. Paremmuus mitataan laskemalla kohdefunktion arvot näille yritteille. Näitä sukupolvia lasketaan yhä uusia, kunnes niiden joukosta löytyy jokin riittävän hyvä ratkaisyksilö. Menetelmä on tehokas ja kohtalaisen helppo toteuttaa valmisohjelmistojen avulla. Vaikeutena on ehkä löytää optimointiongelmaa parhaiten kuvaava kohdefunktio ja sopivat muuttujat, joiden varaan ratkaisusukupolvet rakennetaan.

Korostettakoon tässä, että Bayes-verkot ja geneettiset algoritmit ovat käytännössä aivan eri menetelmiä, joiden sovellusalueetkin ovat erilaiset: Bayes-verkoille päättelymallit, diagnostiikka ja luokittelu, ja puolestaan geneettisille algoritmeille yleiset optimointiongelmat. Ne on kuitenkin tässä kirjassa koottu samoihin kansiin käytännön syistä. Selostukset ovat itsenäiset, ja lukija voi tutustua jompaankumpaan menetelmään aivan toisesta riippumatta.

Erkki Oja

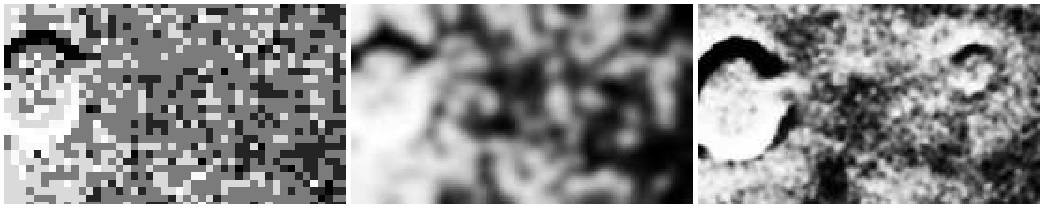
Tiivistelmä

Executive Summary

Sisältö

Johdanto

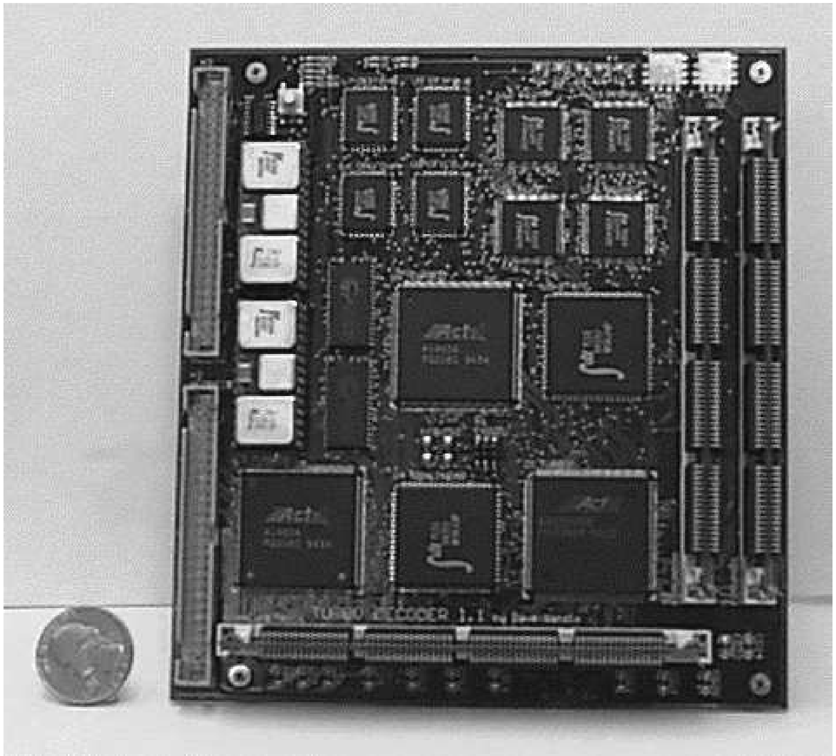
Seuraavassa on lueteltu joukko eri sovellusaloilla (kuvankäsittely, tiedon koodaus ja tiivistys, ohjelmistotekniikka) viime vuosina kehitettyjä merkittäviä teknologioita. Erilaisista sovellusalueistaan huolimatta nämä teknologiat perustuvat samaan taustalla olevaan teoreettiseen kehikkoon.



Kuva 1: Viking-luotaimen Marsin pinnasta ottamista kuvista muodostettu superresoluutiokuva. Vasemmalla: yksi 24 alkuperäisestä kuvasta. Keskellä: kuva perinteisillä kohinansuodatustekniikoilla käsiteltynä. Oikealla: 24 kuvan aineistosta yhdistetty superresoluutiokuva.

Superresoluutio. Viking-avaruusluotain tuottaa useita kuvia Mars-planeetasta. Kuvat on otettu samankaltaisissa valaistusolosuhteissa ja samoista kohteista. Tällöin instrumenttien pienet orientaatio- ja kohdennuserot tuottavat kuitenkin hieman erilaisia kuvia samasta kohteesta. Tällaisesta kuvajoukosta voidaan yhdistelemällä tuottaa tietokoneella “superresoluutiokuva”: kuva joka on tarkempi kuin millään kohinansuodatustekniikalla yksittäisestä kuvasta saatu kuva (kuva 1). National Aeronautics & Space Administration (NASA) käyttää tätä useaan kuvaan perustuvaa superresoluutiotekniikkaa avaruusluotainten tuottamien kohinaisten kuvien parantamiseen, ja tällaisella kuvankäsittelytekniikalla on merkittäviä sovelluksia mm. lääketieteessä ja biotieteissä.

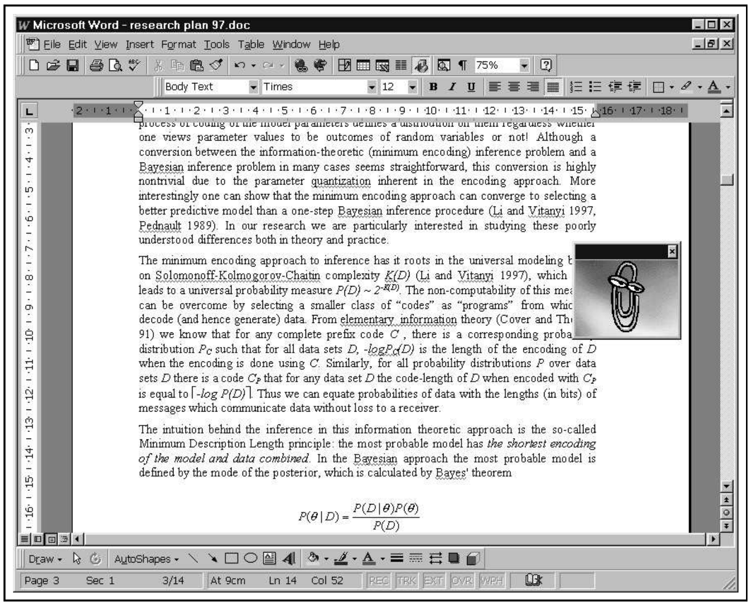
Turbo-koodaus. Tietoliikenteen määrän jatkuvasti kasvaessa tulee tiedon tehokkaasta koodaamisesta ja tiivistämisestä yhä keskeisempiä ongelmia. Vuonna 1993 kehitettyä turbo-koodausta pidetään koodausteorian merkittävimpänä keksintönä vuosikymmeniin. Turbo-koodaus soveltuu tiedon välittä-



Kuva 2: Turbo-koodauksen toteuttava piiri.

miseen kohinaista tiedonsiirtokanavaa käyttäen, ja sen tehokkuus tiedon välityksessä on moninverroin parempi aiempiin menetelmiin verrattuna. Turbokoodit ovat erityisen merkittävä edistysaskel avaruusteknologiassa, jossa ne ylittävät mm. Voyagerin käyttämän ns. "katenointikoodin" (concatenated codes) moninkertaisesti. Turbo-koodaus osoittaa myös kuinka nopeasti teoreettinen tutkimus siirtyy käytännön sovellukseksi, erilaisia turbo-koodeihin perustuvia piirejä valmistavat jo useat yhtiöt kuten Efficient Channel Coding Inc. (kuva 2).

Älykkäät agentit. Nykyiset tekstin ja kuvien käsittelyyn kehitetyt ohjelmistot ovat laajoja ohjelmakokonaisuuksia joiden monien piirteiden tehokas käyttäminen on vaikeaa, ja käytön oppiminen hidasta. Microsoft Office'97-ohjelmistossa käyttäjän apuna ovat "älykkäät agentit" (Office Assistants), jotka itsenäisinä moduuleina tarkkailevat käyttäjän toimintaa ja tarjoavat apuaan käyttäjälle mm. monimutkaisten muotoilujen toteuttamisessa (kuva 3). Tällainen teknologia edellyttää varsin monimutkaista ohjelman toimintojen ja käyttäjän tarpeiden yhteismallinnusta, jotta agentti ei tarjoaisi vain triviaaleja ohjeita. Älykäs agenttiteknologia on erityisesti verkottumisen myö-



Kuva 3: Microsoftin Office Assistant-agentti.

tä muodostumassa yhdeksi tärkeimmistä uusista ohjelmistosuuntauksista, ja sen keskeisimpiä sovellusalueita ovat tiedon haussa avustaminen, liikkuva tietojenkäsittely sekä ohjelmistojen ja laitteiden konfiguroinnissa avustaminen.

Mikä sitten on yhteistä näille edellä lyhyesti esitetyille tekniikoille? Vastaus liittyy sovellusten taustalla olevaan mallinnukseen joka perustuu ns. bayesiläiseen lähestymistapaan muodostaa malleja ja käyttää niitä päätöksenteossa. Yksinkertaisimmillaan bayesiläisen mallinnuksen voi ymmärtää todennäköisyyslaskentaan perustuvana menetelmänä, joka vastaa kysymykseen, miten käsityksemme jostakin asiasta (uskomuksemme jonkin asian totuudesta) muuttuu kun saamme asiaan liittyvää uutta tietoa. Bayesiläinen mallintaminen on kunnioitettavasta iästään huolimatta — perusteet esitettiin jo 200 vuotta sitten — yksi nopeimmin kehittyviä ja kasvavia mallinnusmenetelmiä, ja sen soveltamisella on kauaskantoisia ja merkittäviä vaikutuksia mm. lääketieteessä, teollisessa mallintamisessa, ekonometriassa ja yhteiskuntatieteissä.

Tässä raportissa käsitellään bayesiläistä mallintamista keskittyen erityisesti ns. Bayes-verkkomalleihin, jotka ovat mm. edellä esitettyjen turbo-koodauksen ja Office Assistant-teknologian perustana. Tarkastelemme ensin luvussa 1 bayesiläistä lähestymistapaa yleisesti oppivien ja älykkäiden järjestelmien rakentamisen kannalta, ja sitä kuinka lähestymistapaa voidaan soveltaa mallien konstruoimisessa ja epätäydellisen tiedon varassa toimivassa päättelyssä. Luvussa 2 arvioidaan bayesiläisen mallinnuksen suhdetta muihin oppivien ja älykkäiden järjestelmien yhteydessä esiintyviin menetelmiin kuten esimerkkeihin perustuvaan päättelyyn (Case-Based Reasoning, CBR), ja tarkastellaan lyhyesti bayesiläiseen mallintamiseen liittyvää informaatioteoreettista tulkintaa. Luku 3 tarjoaa katsauksen erilaisiin Bayes-verkkomalleihin ja niihin liittyviin käsitteisiin. Luvussa 4 esitellään Bayes-verkkojen sovelluksia ja Bayes-verkkojen sovellusten kehittämiseen suunniteltuja kaupallisia ohjelmistoja. Lopuksi luvun 5 tavoitteena on esitellä alueen keskeisiä tietolähteitä sekä bayesiläiseen mallintamiseen yleensä että erityisesti Bayes-verkkoihin liittyen.

Luku 1

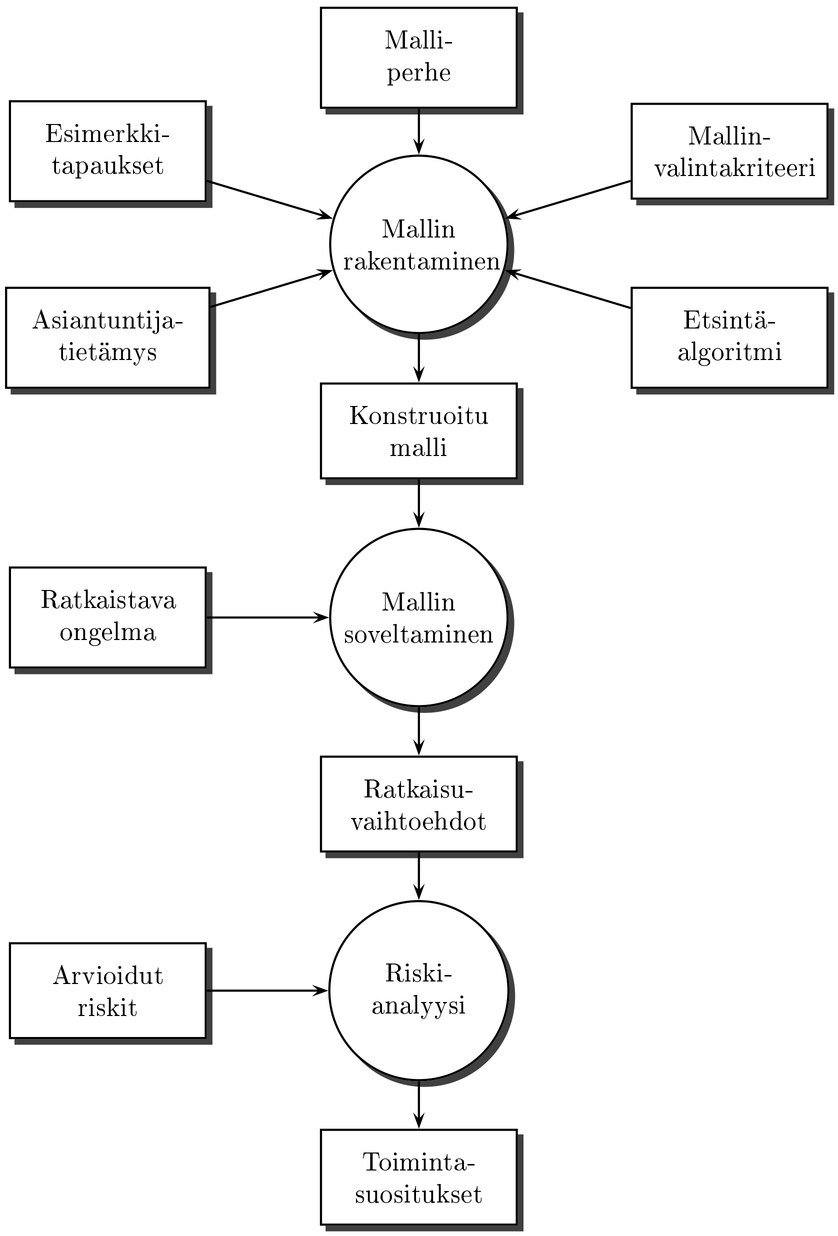
Bayesiläinen mallinnus

Täydellisen tietokoneperustaisen mallin rakentamisen mahdottomuus. Oppivien ja älykkäiden järjestelmien yleinen rakenne — mallien oppiminen ja niiden soveltaminen. Mallien oppiminen bayesiläisittäin: mallien rakenteen ja parametrien oppimisen erottaminen. Ylioppiminen ja Occamin partaveitsi. Oppimisen ja taustatietämyksen yhdistäminen. Bayesiläisten mallien soveltaminen.

1.1 Bayesiläinen mallintaminen oppivissa ja älykkäissä järjestelmissä

Oppivissa ja älykkäissä järjestelmissä pyritään rakentamaan käsiteltävästä sovellusalueesta yksinkertaistettu matemaattinen malli siten, että mallin toteuttavaa tietokoneohjelmaa voidaan käyttää avuksi erilaisissa ongelmanratkaisutilanteissa. Kuvassa 1.1 on tällaisen järjestelmän rakenne jaettu kolmeen komponenttiin: mallien rakentaminen, mallien soveltaminen, ja riskianalyysi. “Mallien soveltaminen” on tässä yhteydessä intuitiivisesti määritelty käsite, joka kattaa mm. seuraavat ongelmakentät:

Tilastollinen päättely ja ennustaminen:
Rakennettu malli on ongelmakentän matemaattinen kuvaus, ja mallinsovellusmoduulille annettua syötettä voidaan tässä tapauksessa ajatella mallinnetussa maailmassa esiintyvän tilanteen osittaisena kuvauksena. Tilastollisessa päättelyssä ja ennustamisessa on päämääränä selvittää mitä annetuista tiedoista seuraa annetun mallin puitteissa, tai mitkä ovat annettuun tilanteeseen johtaneet syyt. Tähän ryhmään kuuluvat muun muassa luokittelu-, diagnosointi-, ja aikasarjaongelmat.



Kuva 1.1: Oppivien ja älykkäiden järjestelmien yleisrakenne.

Ongelmakentästä kerätyn tiedon analysointi (data mining):
Ongelmakentästä kerätystä tiedosta muodostetaan malli, jonka avulla tutkitaan ongelmakentässä esiintyviä yleisiä säännönmukaisuuksia. Mallille annettu syöte (ratkaistava ongelma) määrittelee tässä tapauksessa esimerkiksi millaisista säännönmukaisuuksista käyttäjä on kiinnostunut.

Teollisten prosessien säätö ja ohjaus:
Rakennettu malli kuvaa jotakin todellisen maailman teollista prosessia. Mallille annettu syöte on tässä tapauksessa joukko malliin vaikuttavia parametreja, ja mallin soveltamisessa on tehtävänä päätellä kuinka määrätä loput parametrit optimaalisella tavalla annetun mallin puitteissa.

Tiedon tiivistäminen ja signaalinkäsittely:
Mallinnettu todellisen maailman prosessi käsittelee tässä tapauksessa signaalien tiivistämistä (compression), suodattamista (filtering), tai korjaamista (correction). Mallin sovellusvaiheessa on tehtävänä optimoida tätä prosessia ohjaavat parametrit.

Mikäli mallien sovellusmoduulin antamien tulosten perusteella tehdään toiminnallisia päätöksiä, on riskianalyysivaiheen tehtävänä arvioida eri vaihtoehdoista seuraavat hyödyt ja haitat.

Bayesiläinen mallintaminen on yksi kiinnostava lähestymistapa oppivien ja älykkäiden järjestelmien rakentamiseksi. Lähestymistapa tarjoaa yhtenäisen, todennäköisyyslaskentaan perustuvan formaalin kehikon, jonka avulla rakennettavan järjestelmän eri vaiheissa esiintyvät osaongelmat voidaan muotoilla ristiriidattomalla, teoreettisesti tyydyttävällä tavalla. Termiä “bayesiläinen” käytetään seuraavassa korostamaan kahta mallintamiselle keskeistä seikkaa. Ensinnäkin verrattuna moniin muihin lähestymistapoihin, bayesiläisen mallintamisen lähtökohtana on konstruoida probabilistinen malli ongelmakentän *yhteistodennäköisyysjakaumalle*, siis malli ongelmakentässä esiintyville tilanteille kokonaisuutena, eikä pelkästään tietylle ongelmakentän osalle (esimerkiksi yhden muuttujan todennäköisyysjakaumalle). Yhteistodennäköisyysjakaumaa voidaan luonnollisesti edelleen soveltaa vain tietyn ongelmakentän osa-alueen jakauman estimointiin. Toiseksi termi viittaa myös siihen, että lähestymistavassa käytetään todennäköisyyksien semanttisessa tulkinnassa modernia, *subjektivistista* näkökantaa, jonka mukaan todennäköisyys on subjektiivinen epävarmuuden mitta, eikä perinteisen, ns. *objektivistisen* lähestymistavan mukainen “toistokoefrekvenssi” (tätä seikkaa käsitellään tarkemmin luvussa 2.2). Todennäköisyyslaskennan käyttämisessä ei tietenkään sinänsä ole mitään subjektiivista, vaan teoria johtaa yhteen tiettyyn ristiriidattomaan l. konsistenttiin tapaan käsitellä epävarmaa tietoa. Subjektivistinen lähestymistapa hyväksyy kuitenkin sen tosiseikan, että mikäli päättelyn lähtökohtana käytetään ihmisten arvioimia todennäköisyyksiä, saattavat eri ihmisten antamat arviot tapahtumien todennäköisyyksistä vaihdella hyvinkin paljon, mikä puolestaan vaikuttaa laskennan lopputulokseen. Yksi käytettävän teoreettisen formalismin suurimmista ansioista on se, että kunkin osaongelman ratkaisemisessa käytettävät perusoletukset on luetteloitava eksplisiittisesti, jolloin eri ratkaisujen järkevyyttä voidaan arvioida paitsi empiirisesti, myös analyyttisesti. Seuraavassa käsitellään kutakin järjestelmän kolmesta komponentista erikseen.

1.2 Mallien rakentaminen

Mallien rakentamista voidaan ajatella yleisesti etsintäprosessina, jossa tavoitteena on löytää ongelmakenttää hyvin kuvaavia numeerisia malleja. Etsintäavaruuden muodostava mallien joukko, *malliperhe*, määritellään yleensä käyttäen ns. parametrisia malliperheitä[1]: yksittäistä mallia voidaan tässä lähestymistavassa ajatella parina (M, θ), missä M määrittelee mallin *struktuurin*, esimerkiksi neuroverkon rakenteen tai sumeiden sääntöjen määrän ja muodon, kun taas θ kiinnittää struktuuriin liittyvät parametrit, esimerkiksi neuroverkon kaarten painot tai sumeiden sääntöjen jäsenyysasteet. *Probabilistisella mallilla* tarkoitetaan seuraavassa mallia joka tuottaa todennäköisyysjakauman mallimaailmassa esiintyville tilanteille.

Käytännössä mallien rakentamisessa joudutaan aina rajoittumaan äärelliseen määrään mallistruktuureja, esimerkiksi joukkoon kolmikerroksisia suunnattuja backpropagation-neuroverkkoarkkitehtuureja, joissa piilosolmujen lukumäärä vaihtelee välillä 1–K. Tässä tapauksessa käytettävissä olevia mallistruktuureja on K kappaletta: $M_1, \ldots, M_K$. Malliperheen voidaan nyt ajatella muodostuvan siitä joukosta malleja (M_k, θ), jotka saadaan aikaan käyttämällä mallistruktuurijoukkoa $\{M_1, \ldots, M_K\}$. Tässä yksittäisen neuroverkon kaarien painoja on merkitty (vektorilla) θ. Mallistruktuuria M_k vastaavaksi *malliluokaksi* kutsutaan sitä malliperheen osajoukkoa, joka sisältää kaikkia muotoa (M_k, θ) olevat mallit. Tyypillisiä esimerkkejä oppivissa ja älykkäissä järjestelmissä käytetyistä mallistruktuureista ovat erilaiset neuroverkkorakenteet, päätöspuut, sumeat sääntöjoukot, ja Bayes-verkot, joita käsitellään tarkemmin luvussa 3.

Bayesiläisessä lähestymistavassa mallien rakentamisprosessissa erotetaan eksplisiittisesti *mallinvalintakriteeri*, jonka perusteella päätetään mitkä mallit ovat sopivia käytettäviksi eri ongelmakentissä, ja *etsintäalgoritmi*, jonka avulla etsintäavaruudesta yritetään löytää mallinvalintakriteerin mielessä hyviä malleja. Mikäli käytetty menetelmä mallien konstruoimiseksi ei tuota toivotun kaltaisia tuloksia, voidaan menetelmän evaluoinnissa tämän erottelun ansiosta keskittyä tarkastelemaan erikseen joko käytettyä valintakriteeriä tai etsintäalgoritmia. Esimerkiksi useissa päätöspuualgoritmeissa ei tämänlaa-

[1] Ei-parametrisia malleja ja esimerkkeihin perustuvaa päättelyä (case-based reasoning) käsitellään luvussa 2.3.

Mallistruktuuri: Rakenne, joka määrää mallien määrittelemiseen tarvittavat osat (**p**arametrit). Esimerkiksi neljä syötesolmua, kaksi piilosolmua, ja neljä tulossolmua sisältävä täydellisesti kytketty neuroverkkoarkkitehtuuri muodostaa yhden mallistruktuurin. Yksittäisen mallin määrittelevät parametrit ovat tässä tapauksessa verkon kaarien painot.

Malli: Mallistruktuuri + siihen liittyvien parametrien ilmentymät l. parametrien kiinnitetyt arvot (esimerkiksi $4\times2\times4$–neuroverkkostruktuuri ja siihen liittyvien kaarten painot).

Malliperhe: Joukko mallistruktuureja (esimerkiksi kaikki muotoa $4 \times X \times 4$ olevat neuroverkot, joissa on neljä syötesolmua ja neljä tulossolmua, ja yksi piilokerros). Malliperhettä voidaan ajatella myös kaikkien kiinnitetyllä mallistruktuurijoukolla aikaansaatavien mallien muodostamana joukkona.

Opetusjoukko: Ongelmakentästä saatu otos esimerkkitapauksia.

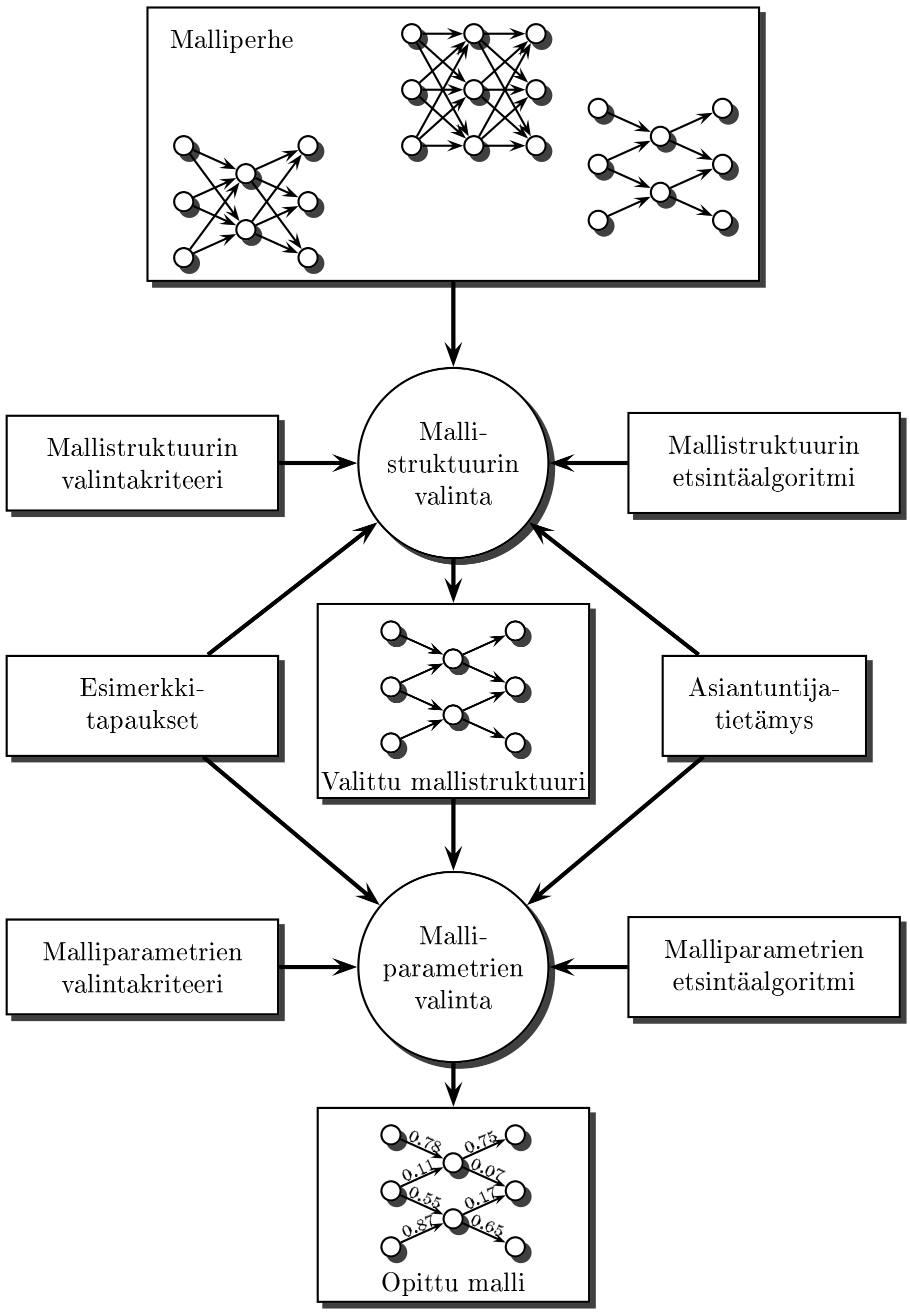
Oppiminen: Mallin (siis mallistruktuurin ja malliparametrien arvojen) valinta kiinnitetystä malliperheestä annettua opetusjoukkoa hyväksikäyttäen.

tuista erottelua ole mahdollista tehdä, vaan etsintä ja valintakriteeri muodostavat yhtenäisen kokonaisuuden, jolloin algoritmien heikkouksien analysointi on vaikeaa.

Kuvassa 1.1 esitetyssä järjestelmässä mallien konstruoiminen tapahtuu käyttäen ongelmankentästä olevaa (asiantuntija)tietämystä, ja/tai saatavilla olevaa ns. *opetusjoukon* (training set) muodostavaa esimerkkiaineistoa. Oletetaan seuraavassa aluksi yksinkertaisuuden vuoksi että käytettävä malli halutaan konstruoida pelkästään opetusjoukon perusteella, jolloin mallien rakennusprosessia kutsutaan mallien *oppimiseksi*, ja käytetään kyseisestä otosjoukosta merkintää $\mathcal{D}$. Asiantuntijatietämyksen yhdistämistä oppimisprosessiin käsitellään luvussa 1.2.4.

Bayesiläisessä mallintamisessa mallien oppiminen voidaan jakaa kahteen erilliseen vaiheeseen: ensimmäisessä vaiheessa määrätään mallistruktuuri M, ja toisessa vaiheessa määrätään kiinnitetyn mallistruktuurin M parametrit θ. Kuvassa 1.2 esitetään tarkennettu kaavio mallien oppimisprosessissta.

Keskeinen ongelma mallien konstruoimisessa on mallien ylisovittamisen välttäminen, jota käsitellään tarkemmin seuraavassa luvussa. Luvuissa 1.2.2



Kuva 1.2: Tarkennettu kuva mallien rakennusvaiheesta.

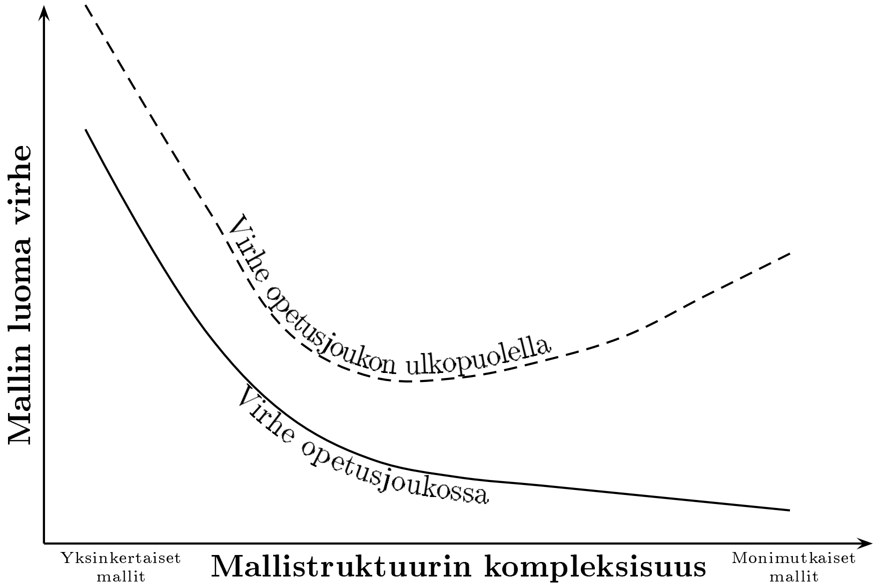
ja 1.2.3 kuvataan bayesiläinen kriteeri mallistruktuurin ja malliparametrien valitsemiseksi. *Tässä yhteydessä on huomattava että bayesiläinen mallintaminen ei ota kantaa oppimisessa käytettävään etsintäalgoritmiin*, joka voi olla mikä tahansa. Yhden vaihtoehdon tällaiseksi algoritmiksi muodostavat *geneettiset algoritmit*, joita käsitellään tarkemmin erillisessä TEKESin teknologiaraportissa.

1.2.1 Mallien ylisovittaminen ja Occamin partaveitsi

Kuten Rissanen "Älykkäiden ja oppivien järjestelmien sovellukset"–teknologiaohjelman evaluointiraportissaan [113] toteaa, valitettavan suuri osa oppiviin ja älykkäisiin järjestelmiin liittyvästä tutkimuksesta on keskittynyt nimenomaan mallien parametrien estimointiongelmaan, vaikka mallin struktuurin valitseminen on käytännön kannalta paljon merkittävämpi ongelma: rakenteellisesti liian monimutkaisissa malleissa on parametrien lukumäärä niin suuri että mallit voidaan *ylisovittaa* (overfit) annettuun opetusjoukkoon. Ylisovittamisen tuloksena syntyvät *ylioppineet* (overtrained) mallit adaptoituvat liian tarkasti opetusjoukkoon, jolloin mallien *yleistyskyky* (generalization capability) eli ennustustarkkuus opetusjoukon ulkopuolisissa tilanteissa on huono. Sanomattakin on selvää, että tällainen järjestelmä, joka toimii hyvin vain tietyissä, ennalta kiinnitetyissä tapauksissa on käytännön sovellusten kannalta hyödytön.

Ylioppiminen: Mallin sovittaminen opetusjoukkoon niin tarkasti, että mallin käyttökelpoisuus on hyvä vain opetusjoukkoon kuuluvissa tapauksissa, ei opetusjoukon ulkopuolella.

Yksi syy ylioppimisilmiöön on se että opetusjoukko on vain suppea ja mahdollisesti virheitä sisältävä otos ongelmakentästä, eikä siten edusta ongelmakentän yhteistodennäköisyysjakaumaa täydellisen tarkasti. Siten opetusjoukkoa vastaavan yhteistodennäköisyysjakauman pikkutarkka estimointi johtaa malliin, joka edustaa huonosti ongelmakentän todellista todennäköisyysjakaumaa. Kuvassa 1.3 on esitetty tyypillinen esimerkki ylioppimisesta: opetusaineistosta malliparametrien sovittamisen tuloksena opitun mallin ennustustarkkuus opetusjoukossa (ennustustarkkuus testattuna käyttäen vain opetusjoukossa annettuja esimerkkitapauksia) on sitä parempi mitä monimutkaisempi käytetty mallistruktuuri on (mitä enemmän mallissa on sovitettavia parametreja), kun taas opitun mallin yleistyskyky eli ennustustarkkuus opetusjoukon ulkopuolisissa tapauksissa alkaa huonontua jossain vaiheessa mallin monimutkaisuuden lisääntyessä. Koska mallien oppimisessa on tavoitteena nimenomaan hyvän yleistyskyvyn saavuttaminen, on ylioppimisen välttäminen koneoppimisen keskeisimpiä ongelmia.



Kuva 1.3: Esimerkki ylioppimisesta.

Ylioppimista voidaan yrittää välttää noudattamalla *Occamin partaveitseksi* kutsuttua periaatetta, jonka mukaan paras mahdollinen malli on sellainen joka sopii ongelmakentästä saatuihin havaintoihin mahdollisimman tarkasti, mutta jonka struktuuri on toisaalta mahdollisimman yksinkertainen. Mallistruktuurin valinnassa on siis löydettävä sopiva tasapaino mallin rakenteen kompleksisuuden ja sen esitysvoimakkuuden välillä: monimutkaiset mallit saadaan sovitettua havaintoihin hyvin tarkasti, mutta niiden struktuurit ovat kompleksisisia, yksinkertaisia malleja ei puolestaan voida useinkaan sovittaa tehtyihin havaintoihin kovin tarkasti. Tilastollisessa oppimisessa tätä tasapainotusongelmaa kutsutaan nimellä *vinouma-varianssi-ongelma (bias-variance dilemma)* (katso esim. [15, 36]).

Vinouma-varianssi-ongelman ratkaisemiseksi on esitetty useita menetelmiä, joista useimmat perustuvat empiirisin argumentein perusteltuihin heuristisiin algoritmeihin (katso esim. [44, 45]). Ongelman ratkaisemiksi on myös esitetty monimutkaisia teoreettisia kehikoita mm. ristiintestauksen (cross-validation) [4] ja VC-ulottuvuuden [137] käsitteisiin perustuen. Esimerkiksi monet neuroverkoissa käytetyt regularisointitekniikat (regularization) [132, 107] pyrkivät ratkaisemaan ylioppisongelman lisäämällä mallinvalintakriteeriin “rankaisutermin”, joka kasvaa neuroverkon rakenteen monimutkaistuessa. Seuraavassa luvussa näemme kuinka bayesiläinen lähestymistapa sisältää “sisäänrakennetun” *automaattisen Occamin partaveitsi-periaatteen*, minkä ansiosta ylioppimisongelmaa ei tarvitse käsitellä bayesiläisessä lähestymistavassa erikseen, vaan se ratkeaa luonnollisena osana mallien konstruointiprosessia!

1.2.2 Mallistruktuurin valinta

Edellä esitetyssä kaksivaiheisessa oppimisprosessissa tarvitsemme oppimiskriteerin sekä mallistruktuurien että mallien parametrien arvojen evaluoimiseksi. Bayesiläisessä oppimisessa käytettävä kriteeri perustuu todennäköisyyteen: mallistruktuuri M_1 on “parempi” kuin mallistruktuuri M_2 jos se on näistä vaihtoehdoista *todennäköisempi*. Tämä periaate pätee sekä mallistruktuureille että mallien parametreille: mallistruktuurit ja malliparametrit evaluoidaan niiden *posterioritodennäköisyyden* mukaan. Mallistruktuurin tapauksessa sana ‘posteriori’ viittaa siihen että struktuurin M todennäköisyys $P(M \mid \mathcal{D})$ lasketaan sen jälkeen kun opetusaineisto on nähty, siis *ehdollisena todennäköisyytenä* annettuna opetusjoukko $\mathcal{D}$. Malliparametrien tapauksessa taas käytetään vastaavasti ehdollista jakaumaa $P(\theta \mid \mathcal{D}, M)$.

Todennäköisyyslaskennan perusaksioomia käyttäen on helppo osoittaa että

$$P(M \mid \mathcal{D}) = \frac{P(\mathcal{D} \mid M)P(M)}{P(\mathcal{D})}. \tag{1.1}$$

Tämä kaava tunnetaan *Bayesin teoreemana* jonka esittäjän, Thomas Bayesin (1701–1761), mukaan koko tutkimusalue on saanut nimensä.

Koska todennäköisyyttä $P(\mathcal{D})$ voidaan pitää tässä yhteydessä vakiona, voidaan mallistruktuurien bayesiläisenä oppimiskriteerinä käyttää siis tuloa, jonka tekijöinä ovat oppimisjoukon *kokonaisuskottavuus* eli *evidenssi (marginal likelihood, evidence)* $P(\mathcal{D} \mid M)$, ja mallistruktuurin *prioritodennäköisyys* $P(M)$. Tässä yhteydessä on huomattava että kokonaisuskottavuus eroaa tilastotieteestä tutusta uskottavuustermistä $P_{\mathcal{D}}$,

$$P_{\mathcal{D}} = P(\mathcal{D} \mid M, \theta),$$

joka antaa opetusjoukon todennäköisyyden annettuna *täydellinen malli*, siis mallistruktuurin lisäksi myös sitä vastaavien parametrien arvot. Kokonais- eli *marginaaliuskottavuus* saadaan “tavallisesta” uskottavuudesta nimensä mukaisesti *marginalisoimalla* eli *integroimalla* uskottavuustermi yli kaikkien mallistruktuuria M vastaavien mallien, painotettuna parametriasetusten θ prioritodennäköisyydellä:

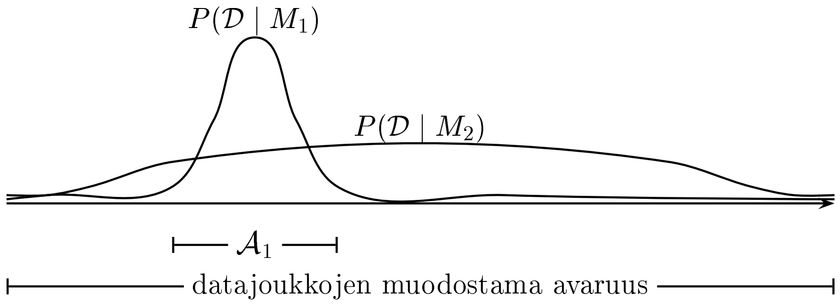
$$P(\mathcal{D} \mid M) = \int P(\mathcal{D} \mid M, \theta) P(\theta \mid M) d\theta. \tag{1.2}$$

Näemme että kokonaisuskottavuutta voidaan pitää normaalin uskottavuustermin odotusarvona niille mallien joukossa joiden struktuuri on M.

Mikäli yllä esiintyvä integraali voidaan laskea, antavat kaavat (1.1) ja (1.2) bayesiläisen kriteerin eri mallistruktuurien vertailemiseksi. Luvussa 3.3 näemme kuinka tämä kriteeri voidaan laskea hyvin tehokkaasti Bayes-verkkojen muodostaman probabilistisen malliperheen tapauksessa. Bayesiläistä lähestymistapaa voidaan soveltaa myös tiettyjen neuroverkkomallien tapauksessa [88, 96, 12, 14, 113], joskaan ei niin suoraviivaisesti kuin probabilististen malliperheiden kanssa.

Yksi tapa soveltaa edellisessä luvussa kuvattua Occamin partaveitsi-periaatetta bayesiläisessä lähestymistavassa saadaan määrittelemällä priorijakauma $P(M)$ siten, että mallistruktuurien prioritodennäköisyys pienenee struktuurin kompleksisuuden kasvaessa. Mikäli mitään muuta syytä — esimerkiksi ongelmakentän asiantuntijan esittämää arvioita mallinnettavan maailman luonteesta — tällaisen priorijakauman käyttämiseen ei kuitenkaan ole, voidaan ei-tasaisen priorijakauman käytöstä luopua, koska *kokonaisuskottavuuskriteeri sisältää jo itsessään automaattisen Occamin partaveitsi-periaatteen!* Tätä aluksi hieman yllättävältä tuntuvaa seikkaa on käsitelty laajemmin mm. teoksissa [110, 88, 96] — asian ymmärtämiseksi intuitiivisella tasolla tarkastelkaamme seuraavaa esimerkkiä.

Olkoon M_1 ja M_2 kaksi mallistruktuuria siten, että M_1 on hyvin yksinkertainen, ja M_2 hyvin monimutkainen mallistruktuuri. Koska siis mallistruktuurin M_2 tapauksessa löytyy aina suuri määrä parametriasetuksia θ, joiden avulla mallistruktuuri voidaan sovittaa tehtyihin havaintoihin $\mathcal{D}$, voimme olettaa että kokonaisuskottavuus $P(\mathcal{D} \mid M_2)$ on kohtalaisen suuri hyvin monelle datajoukolle $\mathcal{D}$. Vastaavasti mallistruktuurin M_1 tapauksessa kokonaisuskottavuus on suuri hyvin rajoitetulle määrälle datajoukkoja; olkoon tämä alue $\mathcal{A}_1$. Koska kokonaisuskottavuus määrää jakauman kaikille mahdollisille datajoukoille $\mathcal{D}$, on selvää että $P(\mathcal{D} \mid M_2)$ ei voi olla suurempi kuin $P(\mathcal{D} \mid M_1)$ kaikilla $\mathcal{D}$. Kuva 1.4 havainnollistaa tätä seikkaa tarkemmin: monimutkaisen mallistruktuurin M_2 tapauksessa on kokonaisuskottavuus levittynyt tasaisesti laajalle, monien datajoukkojen $\mathcal{D}$ muodostamalle alueelle, kun taas mallistruktuurin M_1 tapauksessa kokonaisuskottavuuden merkitsevät arvot saadaan alueella $\mathcal{A}_1$, ja sen ulkopuolella kokonaisuskottavuus on lähellä nollaa. Yksinkertaisempaa mallistruktuuria M_1 vastaava kokonaisuskottavuus on siis suurempi, mikäli käytettävä datajoukko on alueelta $\mathcal{A}_1$!



Kuva 1.4: Intuitiivinen perustelu sille miksi kokonaisuskottavuus noudattaa automaattisesti Occamin partaveitsi-periaatetta: molempien käyrien rajaaman alueen pinta-alan on oltava yhtä suuri (yksi), jolloin yksinkertainen mallistruktuuri M_1 saa jollakin alueella (tässä $\mathcal{A}_1$) suuremman kokonaistodennäköisyyden arvon kuin monimutkaisempi mallistruktuuri M_2.

Occamin partaveitsi-periaatteen toteutumisen näemme myös soveltamalla kokonaisuskottavuuden laskemisessa ns. BIC-approksimaatiota [120, 109, 65], jonka mukaan

$$P(\mathcal{D} \mid M) \approx \frac{P(\mathcal{D} \mid \tilde{\theta}(\mathcal{D}))}{N^{\frac{d(M)}{2}}},$$

missä merkintä $\tilde{\theta}(\mathcal{D})$ tarkoittaa uskottavuuden $P(\mathcal{D}|\theta(\mathcal{D}))$ maksimoivia *suurimman uskottavuuden parametreja (maximum likelihood parameters)*, $\mathrm{d}(M)$ mallistruktuuria M vastaavien malliparametrien lukumäärää, ja N datajoukon $\mathcal{D}$ kokoa. Näemme siis että kokonaisuskottavuus tasapainoilee Occamin partaveitsi-periaatteen mukaisesti mallin esityskyvyn ja kompleksisuuden välillä: mallin struktuurin kompleksisuuden kasvaessa osoittajassa esiintyvä uskottavuustermi kasvaa, mutta samanaikaisesti kasvaa myös nimittäjässä esiintyvä mallin kompleksisuudesta rankaiseva termi.

Jos esitämme BIC-approksimaation hieman toisessa muodossa (ottamalla logaritmit kyseisistä termeistä), saamme

$$-\log P(\mathcal{D} \mid M) \approx -\log P(\mathcal{D} \mid \tilde{\theta}(\mathcal{D})) + \frac{d(M)}{2} \log N,$$

mistä näemme että approksimaatio on sama kuin Rissasen teoksessa [109] esittämä *Minimum Description Length (MDL)*-periaatteeseen perustuva kaksiosainen kriteeri mallistruktuurin valitsemiseksi. Bayesiläisen ja informaatioteoreettisten lähestymistapojen yhteyksiä käsitellään myöhemmin tarkemmin luvussa 2.4.

1.2.3 Malliparametrien valinta

Kun kriteerin (1.1) mielessä optimaalinen mallistruktuuri M on löydetty, on seuraava tehtävä oppimisprosessissa määrätä mallistruktuuriin liittyvät parametrit θ. Monissa ei-probabilistisissa malliperheissä, kuten esimerkiksi neuroverkkomalleissa, käytetty mallinvalintakriteeri on tässä tapauksessa yleensä estimointivirhe opetusjoukossa. Kuten edellä nähtiin, tämä johtaa helposti ylioppimiseen mikäli mallistruktuuria ei ole valittu oikein. Käytetyt etsintäalgoritmit (esimerkiksi backpropagation) ovat näissä tapauksissa yleensä yksinkertaisia iteratiivisia gradienttimenetelmän variaatioita joiden konvergenssi saattaa olla hyvin hidasta. Lisäksi gradienttimenetelmät ovat lokaaleja menetelmiä jotka parhaassakin tapauksessa konvergoivat vain lokaaliin optimiin, eivätkä välttämättä löydä niitä parametriarvoja jotka maksimoivat käytetyn parametrien evaluointikriteerin.

Bayesiläisessä mallintamisessa sitä vastoin parametrien arvojen valinnassa käytettävä oppimiskriteeri on sama kuin mallistruktuurien valinnassa: parametriarvot θ evaluoidaan niiden posterioritodennäköisyyden $P(\theta \mid M, \mathcal{D})$ mukaan. Todennäköisyyslaskennan perusaksioomista saamme että

$$P(\theta \mid M, \mathcal{D}) = \frac{P(\mathcal{D} \mid \theta, M) P(\theta \mid M)}{P(\mathcal{D} \mid M)}, \tag{1.3}$$

Yksittäisen mallin θ posterioritodennäköisyys saadaan siis kertomalla opetusjoukon uskottavuus $P(\mathcal{D} \mid \theta, M)$ mallin prioritodennäköisyydellä $P(\theta \mid M)$, ja normalisoimalla syntyvä tulo kokonaisuskottavuudella (1.2), jonka arvo on jo määrätty mallistruktuurin valinnassa. Tietyissä probabilistissa malliperheissä posteriorijakauman maksimikohta voidaan laskea suljetussa muodossa yksinkertaisella matemaattisella kaavalla, jolloin *mallin parametrien säätämiseen ei tarvita lainkaan iteratiivista aikaavievää prosessia*! Esimerkin tällaisesta malliperheestä muodostavat tässä raportissa käsiteltävät Bayes-verkot, joita tarkastellaan tarkemmin luvussa 3. Luvussa 3.3.2 esitetään yksinkertainen laskukaava, jonka avulla annetun Bayes-verkon parametrit voidaan määrätä suoraan ilman iteratiivista oppimisprosessia.

1.2.4 Oppimisen ja asiantuntijatietämyksen yhdistäminen

Edellä oletettiin mallistruktuurin valinnassa käytettävän pelkästään tilastollista havaintoaineistoa $\mathcal{D}$. Usein saatavilla on kuitenkin esimerkkitapauksien lisäksi myös runsaasti "ihmistietämystä": ongelmakentän asiantuntijoiden kokemusperäistä tietoa, tai "maalaisjärkeen" (common sense) perustuvaa tietoa ongelmakentän luonteesta. Monien malliperheiden, kuten esimerkiksi neuroverkkojen, tapauksessa merkittäväksi ongelmaksi on muodostunut se, kuinka nämä kaksi tietolähdettä saadaan yhdistettyä. Bayesiläisessä mallintamisessa yhdistäminen on suoraviivaista: kaavoissa (1.1) ja (1.2) esiintyy kaksi prioritodennäköisyysjakaumaa, $P(M)$ ja $P(\theta \mid M)$. Sana 'priori' viittaa tässä yhteydessä siihen että nämä jakaumat on määrättävä "ennen" opetusjoukon $\mathcal{D}$ tarkastelua siinä mielessä että ne eivät saa riippua opetusjoukon muodostavasta tilastollisesta aineistosta. Bayesiläinen formalismi antaa nyt mahdollisuuden koodata ongelmakentästä saatavilla olevaa yleistä tietämystä prioritodennäköisyyksien avulla.

Luvussa 3.3 näemme että Bayes-verkkojen tapauksessa mallistruktuurien prioritodennäköisyyksien arvioiminen on erityisen luontevaa, koska kukin Bayes-verkkostruktuuri edustaa joukkoa muuttujien välisiä riippuvuuksia, ja tällaisten riippuvuuksien arviointi on ongelmakentän asiantuntijoille monessa tapauksessa sangen helppoa. Lisäksi priorijakauma $P(\theta \mid M)$ voidaan formalisoida Bayes-verkkojen tapauksessa tavalla, joka antaa asiantuntijoille mahdollisuuden paitsi koodata prioritietämyksensä, myös arvioida prioritiedon merkityksellisyyttä suhteessa saatavilla olevaan tilastolliseen aineistoon. Toisaalta on myös korostettava, että koska Bayes-verkkojen tapauksessa on sekä mallistruktuurilla että siihen liittyvillä parametreilla selvä semanttinen merkitys, voidaan Bayes-verkkoja konstruoida aivan hyvin myös ilman tilastollista aineistoa, käyttäen puhtaasti asiantuntijoilta saatua tietämystä.

1.3 Mallien soveltaminen

Kuten edellä näimme, bayesiläisessä oppimisessa muodostetaan ongelmakentästä malli (M, θ) käyttäen apuna opetusesimerkkejä $\mathcal{D}$, ja/tai ongelmakentän asiantuntijoiden prioritietämystä. Oletetaan jatkossa että ongelmakentän kuvauksessa käytetään n attribuuttia (satunnaismuuttujaa) $X_1, \ldots, X_n$. Merkinnällä $X_i = x_i$ tarkoitetaan sitä että attribuutin X_i arvo on x_i, missä x_i voi olla joko reaalilukuku, jolloin kyseessä on *jatkuva-arvoinen* muuttuja, tai sitten x_i kuuluu johonkin äärelliseen joukkoon mahdollisia arvoja, missä tapauksessa attribuuttia X_i kutsutaan *diskreetiksi.* Jatkossa keskitymme yksinkertaisuuden vuoksi hyvin paljon diskreetteihin muuttujiin — jatkuvien muuttujien käsittelystä puhutaan enemmän luvussa 3.4.

Seuraavassa oletamme että käytetty malliperhe on probabilistinen, missä tapauksessa käytettävä malli (M, θ) määrittelee yhteistodennäköisyysjakauman $P(X_1 = x_1, \ldots, X_n = x_n \mid M, \theta)$ muuttujien $X_1, \ldots, X_n$ eri arvokombinaatioille. Tätä yhteistodennäköisyysjakaumaa käyttäen on mahdollista rat-

Muuttujajoukko $\mathbf{S}_1$:
Ne muuttujat joiden arvo ei ole tiedossa, mutta joiden arvojen tuntemisen katsotaan olevan edellytyksenä käsillä olevan ongelmanratkaisutilanteen ratkaisemiksi.

Muuttujajoukko $\mathbf{S}_2$:
Muuttujat joiden arvo on tunnettu.

Muuttujajoukko $\mathbf{S}_3$:
Muuttujat joiden arvo ei ole tiedossa, eikä arvojen tuntemisen katsota olevan välttämätön käsillä olevan ongelmanratkaisutilanteen kannalta.

Probabilistinen päättely:
Todennäköisyysjakauman $P(\mathbf{S}_1 \mid \mathbf{S}_2 = \mathbf{s}_2, M, \theta)$ estimointi.

kaista erilaisia mallien soveltamisessa esiintyviä estimointitehtäviä. Tätä havainnollistaaksemme olettakaamme että attribuutit on jaettu kolmeen erilliseen osajoukkoon siten että ensimmäisen osajoukon $\mathbf{S}_1$ muodostavat ne attribuutit joiden arvoa halutaan estimoida, toisen osajoukon $\mathbf{S}_2$ ne attribuutit joiden arvo on annettu, ja kolmannen osajoukon $\mathbf{S}_3$ muut attribuutit. *Probabilistisessa päättelyssä* tavoitteena on laskea ehdollinen todennäköisyysjakauma kiinnostuksen kohteena oleville attribuuteille, annettuna tunnettujen attribuuttien arvot. Tuloksena on siis jakauma $P(\mathbf{S}_1 \mid \mathbf{S}_2 = \mathbf{s}_2, M, \theta)$, missä merkintä $\mathbf{S}_2 = \mathbf{s}_2$ tarkoittaa että kukin attribuutti joukossa $\mathbf{S}_2$ on asetettu tunnettuun arvoonsa.

Tässä yhteydessä on korostettava että attribuuttien jako yllämainittuihin kolmeen osajoukkoon ei ole millään tavalla kiinteä, vaan voi vaihdella tilanteen mukaan. Niinpä probabilistisessa päättelyssä ei tarvitse etukäteen päättää mitä halutaan estimoida tai mitä tietoa estimointitehtävässä on saatavilla. Tämä poikkeaa monista vaihtoehtoisista lähestymistavoista, kuten esimerkiksi päätöspuista ja monissa neuroverkkomalleista, joissa estimointitehtävä (esimerkiksi luokittelutehtävä) on päätettävä etukäteen jo ennen mallin rakentamista. Probabilistisessa päättelyssä sitä vastoin riittää, jos yhteistodennäköisyydet $P(X_1, \ldots, X_n \mid M, \theta)$ on annettu: todennäköisyyslaskennan perusaksioomista seuraa että ehdolliset todennäköisyydet voidaan esittää muodossa

$$\begin{aligned} P(\mathbf{S}_1 = \mathbf{s}_1 \mid \mathbf{S}_2 = \mathbf{s}_2, M, \theta) &= \frac{P(\mathbf{S}_1 = \mathbf{s}_1, \mathbf{S}_2 = \mathbf{s}_2 \mid M, \theta)}{P(\mathbf{S}_2 = \mathbf{s}_2 \mid M, \theta)} \\ &= \frac{\sum_{\mathbf{S}_3} P(\mathbf{S}_1 = \mathbf{s}_1, \mathbf{S}_2 = \mathbf{s}_2, \mathbf{S}_3 \mid M, \theta)}{\sum_{\mathbf{S}_1, \mathbf{S}_3} P(\mathbf{S}_1, \mathbf{S}_2 = \mathbf{s}_2, \mathbf{S}_3 \mid M, \theta)}, \end{aligned} \tag{1.4}$$

missä merkintä $\sum_{\mathbf{S}_i}$ tarkoittaa että summassa käydään läpi kaikki joukossa $\mathbf{S}_i$ esiintyvien muuttujien arvokombinaatiot. Mikä tahansa ehdollinen todennäköisyys saadaan siis laskettua summaamalla eli *marginalisoimalla* yli tietyn yhteistodennäköisyyksien joukon. Jotta tätä lähestymistapaa olisi mahdollista soveltaa käytännössä, on ratkastava seuraavat kaksi ongelmaa:

1. Kuinka tallettaa yhteistodennäköisyydet $P(X_1, \ldots, X_n \mid M, \theta)$ niin että niiden talletus ei vaadi eksponentiaalista muistitilaa, ja että ne ovat tehokkaasti käytettävissä?

2. Kuinka laskea kaavassa (1.4) esiintyviä marginaalisummia tehokkaasti?

Luvussa 3 osoitamme kuinka yllä esitetyt ongelmat voidaan ratkaista kun yhteistodennäköisyysjakaumat määritellään käyttäen Bayes-verkkojen malliperhettä: luvussa 3.1 käsitellään yhteistodennäköisyyksien tallettamisen ongelmaa, ja luvussa 3.2 marginalisointiongelmaa Bayes-verkkojen tapauksessa.

Probabilistisessa päättelyssä estimoidaan siis ehdollista jakaumaa $P(\mathbf{S}_1 \mid \mathbf{S}_2 = \mathbf{s}_2, M, \theta)$. Seuraavassa luvussa käsitellään sitä miten tätä jakaumaa voidaan käyttää hyväksi päätöksenteossa arvioimalla eri toimenpiteistä seuraavien hyötyjen ja haittojen odotusarvoja. Ennen sitä kuitenkin esittelemme joitakin tyypillisiä probabilistisen päättelyn muotoja.

Yksinkertaisimmassa tapauksessa joukko $\mathbf{S}_1$ käsittää vain yhden muuttujan X, ja tehtävänä on määrätä muuttujan X arvojen jakauma lähtötietojen $\mathbf{S}_2 = \mathbf{s}_2$ pohjalta. Tyypillisen esimerkin tällaisesta ongelmasta muodostavat *luokitteluongelmat*, joissa tehtävänä on määrittää diskreetin luokkamuuttujan X arvojen todennäköisyydet (katso esimerkki 1).

Esimerkissä 1 käytettyjen diskreettien muuttujien jakauma voidaan määrätä laskemalla todennäköisyys kullekin muuttujan mahdolliselle arvolle. Mikäli X on jatkuva-arvoinen, jakauma voidaan tietyin jakaumaoletuksin kuvata antamalla jakauman tunnusluvut, esimerkiksi odotusarvo ja varianssi (katso esimerkki 2).

Esimerkissä 1 esitetyissä diskreeteissä luokitteluongelmissa probabilistista päättelyä käytettiin diagnostiseen päättelyyn, eli syiden (esim. sairaus) todennäköisyyksien arviointiin, annettuna havaitut seuraukset (oireet). Toisaalta, kuten edellä mainittiin, muuttujien jakoa joukkohin $\mathbf{S}_1$, $\mathbf{S}_2$ ja $\mathbf{S}_3$ voi-

Esimerkki 1 Yhden diskreetin muuttujan jakauman estimointi (luokitteluongelma).

Ongelma1:	Sairastaako potilas tautia X havaittujen oireiden ja tehtyjen kokeiden ($\mathbf{s}_2$) perusteella?
Ongelma2:	Mikä on asiakkaan luottokelpoisuusluokka (X) havaitun osittaisen asiakasprofiilin ($\mathbf{s}_2$) perusteella?
Ratkaisu1:	Laske $P(X = 1 \mid \mathbf{S}_2 = \mathbf{s}_2, M, \theta)$.
Ratkaisu2:	Laske $P(X = x \mid \mathbf{S}_2 = \mathbf{s}_2, M, \theta)$ kaikille luottokelpoisuusluokille x.

Esimerkki 2 Jatkuvan muuttujan jakauman estimointi.

Ongelma1:	Mikä on tuotteen valmistuskustannusten (X) oletettava arvo, jos täytämme asiakkaan tuotteelle asettamat vaatimukset ($\mathbf{s}_2$)?
Ongelma2:	Mikä on ulostuloventtiilin X virtauksen oletettava voimakkuus jos säätimet $\mathbf{S}_2$ asetetaan arvoihin $\mathbf{s}_2$?
Ratkaisu:	Laske jakauman $P(X \mid \mathbf{S}_2 = \mathbf{s}_2, M, \theta)$ odotusarvo.

daan muuttaa dynaamisesti mielivaltaisella tavalla. Niinpä siirtämällä muuttujia joukosta toiseen voimme soveltaa probabilistista päättelyä myös päinvastaiseen suuntaan, seurausten ennustamiseen syiden pohjalta. Tämänkaltaista päättelyä voidaan käyttää esimerkiksi ongelmakentän luonteen analysointiin (data mining) esimerkeissä 3 esitetyillä tavoilla.

Esimerkki 3 Ongelmakentän analysointi.

Ongelma1:	Kuinka suuri osa tautia X sairastavista potilaista kärsii äkillisistä tajuttomuuskohtauksista ($Y = 1$)?
Ongelma2:	Mikä on parhaaseen luottokelpoisuusluokkaan (AAA) kuuluvien asiakkaiden yleisin tilimuoto (Y)?
Ratkaisu1:	Laske $P(Y = 1 \mid X = 1, M, \theta)$.
Ratkaisu2:	Laske $P(Y = y \mid X = AAA, M, \theta)$ kaikille tilimuodoille y.

Esitettyjä kahta päättelyn muotoa (seurauksien arviointi syistä, tai syiden päättely seurauksista) voidaan tietenkin edelleen sekoittaa mielivaltaisella tavalla ratkaisemaan mm. esimerkissä 4 esitettyjen tapausten kaltaisia ongelmia.

Mikäli joukko $\mathbf{S}_1$ sisältää useita muuttujia, tulee jakauman $P(\mathbf{S}_1 \mid \mathbf{S}_2 = \mathbf{s}_2, M, \theta)$ täydellisestä määräämisestä vaikeaa, koska se vaatii diskreetissäkin tapauksessa kaikkien joukossa $\mathbf{S}_1$ olevien muuttujien arvokombinaatioiden lä-

Esimerkki 4 Eri päättelytapojen yhdistäminen.

Ongelma1:	Onko oletettavaa että potilas voi kärsiä tajuttomuuskohtauksista ($Y = 1$), jos hänellä on tauti X, ja hänen ruumiinlämpötilansa on z?
Ongelma2:	Onko tehtyjen havaintojen ($\mathbf{s}_2$) perusteella oletettavaa että potilas sairastaa tautia X, ja että hän voi saada vakavia tajuttomuuskohtauksia?
Ratkaisu1:	Laske $P(Y = 1 \mid X = 1, Z = z, M, \theta)$.
Ratkaisu2:	Laske $P(X = 1, Y = 1 \mid \mathbf{S}_2 = \mathbf{s}_2, M, \theta)$.

pikäynnin, ja näitä arvokombinaatioita on tietenkin eksponentiaalinen määrä. Niinpä tällaisessa tilanteessa tyydytäänkin usein etsimään se muuttujien $\mathbf{S}_1$ arvokombinaatio $\mathbf{s}_1$ (tai n tässä mielessä parasta arvokombinaatiota) jolla on korkein todennäköisyys, ts. arvokombinaation joka maksimoi todennäköisyyden $P(\mathbf{S}_1 = \mathbf{s}_1 \mid \mathbf{S}_2 = \mathbf{s}_2, M, \theta)$. Koska

$$\max_{\mathbf{s}_1} P(\mathbf{S}_1 = \mathbf{s}_1 \mid \mathbf{S}_2 = \mathbf{s}_2, M, \theta) = \max_{\mathbf{s}_1} P(\mathbf{S}_1 = \mathbf{s}_1, \mathbf{S}_2 = \mathbf{s}_2 \mid M, \theta), \tag{1.5}$$

näemme että tässä mielessä optimaalinen arvokombinaatio täydentää vapaille muuttujille $\mathbf{S}_1$ arvot siten että syntyvän kokonaisarvokombinaation todennäköisyys on maksimaalinen. Tyypillisiä esimerkkejä tämänkaltaisen probabilistisen päättelyn sovellusalueista ovat konfiguraatio-ongelmat, joissa täydennetään osakonfiguraatio siten että täydennetyn konfiguraation todennäköisyys on mahdollisimman suuri annetun mallin muodostamassa todennäköisyysjakaumassa. Jos malli (M, θ) muodostetaan esimerkiksi jotakin tuotetta valmistavan yrityksen valmistettujen tuotteiden tietokannasta, voidaan tällä tavalla täydentää asiakkaan tuotteelle esittämät toivomukset siten että uusi tuote on mahdollisimman samankaltainen jo valmistettujen tuotteiden kanssa, ja siten otaksuttavasti edullinen valmistaa (katso esimerkki 5).

Esimerkki 5 Optimaalisen konfiguraation etsiminen.

Ongelma1:	Täydennä valmistettavan tuotteen kuvaus ($\mathbf{S}_1$) siten että tuote täyttää asiakkaan sille esittämät vaatimukset ($\mathbf{s}_2$), ja sopii yrityksen tuoteprofiiliin (eli siitä tehtyyn malliin) mahdollisimman hyvin.
Ongelma2:	Konstruoi asiakkaan täydellinen asiakasprofiili ($\mathbf{S}_1$) annettujen tietojen $\mathbf{s}_2$ pohjalta.
Ratkaisu:	Etsi $\mathbf{s}_1$ siten että todennäköisyyden $P(\mathbf{S}_1 = \mathbf{s}_1, \mathbf{S}_2 = \mathbf{s}_2 \mid M, \theta)$ arvo on maksimaalinen.

Ehdollisia todennäköisyyksiä voidaan käyttää myös etäisyysmittana, jonka avulla voidaan etsiä esimerkiksi valmistettujen tuotteiden joukosta se tapaus joka muistuttaa eniten asiakkaan antamaa osittaista tuotteen kuvausta. Tällaista probabilistista etäisyysmetriikkaa käsitellään lähteessä [70].

Konfiguraatio-ongelmia läheisesti muistuttavan sovellusalueen muodostavat asiakasprofilointitehtävät (katso esimerkki 5). Tavoitteena on täydentää asiakkaan profiili asiakkaasta kerättyjen tietojen ja kaikkien asiakkaiden muodostamasta tietokannasta muodostetun mallin perusteella. Syntyvää täydennettyä asiakasprofiilia voidaan käyttää hyväksi mm. suoramarkkinoinnin suuntaamisessa: probabilistista päättelyä voidaan käyttää esimerkiksi ennustamaan olisiko asiakas kiinnostunut tietyntyyppisestä luottokortista annettuna hänestä kerätyt tiedot.

Koska probabilistista päättelyä käyttäen voidaan muodostaa täydellinen mallimaailman tilanteen kuvaus, voidaan yhteistodennäköisyyden maksimoivia konfiguraatioita käyttää myös probabilistisen päättelyn selitysmekanismina: muodostetusta täydennetystä konfiguraatiosta voidaan esimerkiksi päätellä mitä muita tauteja tai oireita jo diagnosoidulla potilaalla mahdollisesti on (katso esimerkki 6). Ennustettujen lisäoireiden havaitseminen lisää luonnollisesti jo tehdyn diagnoosin luotettavuutta.

Esimerkki 6 Päättelyä tukevat selitykset.

Ongelma:	Oletetaan että potilas sairastaa tautia X ja että hänellä on oireet $\mathbf{S}_2 = \mathbf{s}_2$. Mitä muita oireita potilaalla luultavasti on? Sairastaako potilas mahdollisesti myös jotain muuta tautia?
Ratkaisu:	Etsi $\mathbf{s}_3$ siten että todennäköisyyden $P(X = 1, \mathbf{S}_2 = \mathbf{s}_2, \mathbf{S}_3 = \mathbf{s}_3 \mid M, \theta)$ arvo on maksimaalinen.

Kuten edellä esitetyistä esimerkeistä huomaamme, puuttuva tieto käsitellään probabilistisessa päättelyssä erityisen elegantilla tavalla: jos esimerkiksi alkuperäinen suunnitelmamme oli estimoida jakaumaa $P(X_1 \mid X_2 = x_2, \ldots, X_5 = x_5)$, mutta muuttujan X_3 arvoa ei olekaan saatavilla, käytämme päättelyssä yksinkertaisesti jakaumaa $P(X_1 \mid X_2 = x_2, X_4 = x_4, X_5 = x_5)$ — kullakin hetkellä käytetään siis jakaumaa joka saadaan ottamalla huomioon kaikki annettu tieto. Monien muiden malliperheiden tapauksessa on puuttuvan tiedon käsittely huomattavan hankalaa: esimerkiksi päätöspuiden toimivuuden perusedellytyksenä on se, että kussakin päätössolmussa käytettävän muuttujan arvo on aina mahdollista saada selville.

Probabilistista päättelyä voidaan käyttää myös puuttuvan tiedon estimointiin: sen sijaan että estimoisimme jakaumaa $P(\mathbf{S}_1 \mid \mathbf{S}_2 = \mathbf{s}_2, M, \theta)$, asetamme estimoitavat muuttujat johonkin arvokombinaatioon $\mathbf{s}_1$, ja kysymme kuinka muuttujat $\mathbf{S}_3$ on asetettava kun halutaan maksimoida halutun arvokombinaation todennäköisyys $P(\mathbf{S}_1 = \mathbf{s}_1 \mid \mathbf{S}_2 = \mathbf{s}_2, \mathbf{S}_3, M, \theta)$. Esimerkissä 7 esitetyt tapaukset kuvaavat tämäntyyppisiä sovellusalueita.

Esimerkki 7 Puuttuvan tiedon estimointi.

Ongelma1:	Mitä lääkettä ($\mathbf{S}_3$) potilaalle on annettava jotta toipumisen todennäköisyys olisi maksimaalinen?
Ongelma2:	Kuinka asettaa säätimet $\mathbf{S}_3$ siten että läpivirtaus putkessa X olisi optimaalinen tilanteessa jossa säätimet $\mathbf{S}_2$ halutaan pitää nykyisissä asetuksissa $\mathbf{s}_2$?
Ongelma3:	Muodosta valmistettavan tuotteen kuvaus ($\mathbf{S}_3$) siten että se on konsistentti asiakkaan vaatimusten ($\mathbf{s}_2$) kanssa, ja voitto on maksimaalinen.
Ratkaisut1&2:	Etsi $\mathbf{s}_3$ siten että todennäköisyyden $P(X = 1 \mid \mathbf{S}_2 = \mathbf{s}_2, \mathbf{S}_3 = \mathbf{s}_3, M, \theta)$ arvo on maksimaalinen.
Ratkaisu3:	Etsi $\mathbf{s}_3$ siten että jakauman $P(X \mid \mathbf{S}_2 = \mathbf{s}_2, \mathbf{S}_3 = \mathbf{s}_3 \mid M, \theta)$ odotusarvo on maksimaalinen.

Tässä yhteydessä on huomautettava että esimerkissä 7 esiintyvä päättelyn muoto, puuttuvan tiedon estimointi, on monissa tilanteissa laskennallisesti raskaampi tehtävä kuin jakauman $P(\mathbf{S}_1 \mid \mathbf{S}_2 = \mathbf{s}_2, M, \theta)$ estimointi, eli päättely puuttuvan tiedon vallitessa.

Paitsi probabilistisessa päättelyssä esiintyvissä syötearvoissa, puuttuvaa tietoa voi esiintyä myös opetusaineistossa $\mathcal{D}$. Lähteessä [72] tarkastellaan bayesiläisen lähestymistavan käyttämistä tämän ongelman ratkaisemisessa.

1.4 Riskianalyysi

Edellisessä luvussa kuvattiin kuinka probabilistisessa päättelyssä on tavoitteena muodostaa ehdollinen todennäköisyysjakauma estimoitaville muuttujille, annettuna tunnettujen muuttujien arvot. Oletetaan seuraavassa yksinkertaisuuden vuoksi että estimoimme jakaumaa $P(X \mid \mathbf{S}_2 = \mathbf{s}_2, \theta, M)$, missä X on diskreetti muuttuja jonka arvo meidän on kiinnitettävä annettujen lähtötietojen $\mathbf{s}_2$ pohjalta. Intuitiivisesti luonnollisimmalta tuntuvin vaihtoehto on tietenkin valita se muuttujan X arvo x joka maksimoi todennäköisyyden $P(X = x \mid \mathbf{S}_2 = \mathbf{s}_2, \theta, M)$. Käytännön sovelluksissa saatetaan kuitenkin tehdä muuttujan arvon valinnan perusteella päätöksiä, joiden seuraukset voivat olla hyvin kauaskantoisia, ja siksi päätöksenteon huolellinen analyysi on paikallaan. Todennäköisyyslaskenta tarjoaa *päätösteoriana (decision theory)* tunnetun teoreettisen kehikon eri päätöstilanteissa esiintyvien ongelmien ratkaisemiseksi. Päätösteorian mukaan meidän on valittava se vaihtoehto, joka minimoi odotettavissa olevan haitan (maksimoi odotettavissa olevan hyödyn). Esimerkki 8 havainnollistaa tätä periaatetta.

Esimerkki 8 Päätösteoreettinen riskianalyysi.

Työpaikkalääkäri saa eteensä työntekijän, jolle tehdyt alustavat testit osoittavat henkilön potevan erästä tautia 60% todennäköisyydellä. Tauti on sinänsä vaaraton, mutta jos se pääse hoitamattomana taudin myöhempään vaiheeseen, joutuu henkilö kymmenen päivän sairaslomalle. Tauti voidaan diagnosoida 100% luotettavuudella erään nopean, mutta melko kalliin lisätestin avulla. Testin hinta vastaa työntekijän yhden päivän työpanosta. Jos henkilöllä tämän testin avulla havaitaan kyseinen tauti ennen kuin se kehittyy myöhempään vaiheeseensa, voidaan tarvittavien sairaspäivien määrä vähentää puoleen.

Lääkäri pohtii tulisiko hänen määrätä työntekijälle kallis lisätesti, vai luokitella henkilö terveeksi ja lähettää hänet takaisin töihin. Eri vaihtoehdoissa kustannukset muodostuvat siis seuraavasti, kun yksikkönä käytetään menetettyjä työpäiviä:

Aiheutuvat kustannukset:	Potilas on sairas	Potilas on terve
Tehdään lisätesti	6	1
Ei tehdä lisätestiä	10	0

Jos potilaalle tehdään lisätesti, on aiheutuvien kustannusten odotusarvo $0.6 \cdot 6 + 0.4 \cdot 1 = 4$, kun taas toisessa tapauksessa aiheutuvien kustannusten odotusarvo on $0.6 \cdot 10 + 0.4 \cdot 0 = 6$. Päätösteorian mukaan eri toimintavaihtoehdoista on valittava se, josta seuraava odotettavissa oleva kustannus on minimaalinen (tai vastaavasti odotettavissa oleva hyöty maksimaalinen), joten lääkärin tulisi määrätä potilas lisätesteihin. Jos lääkärille tulee 1000 samanlaista tapausta vuodessa, toimimalla päätösteorian mukaisesti hän säästää yhtiölleen vuosittain (keskimäärin) 2000 työpäivää vastaavan määrän rahaa.

Jos ehdolliset todennäköisyydet $P(X = x \mid \mathbf{S}_2 = \mathbf{s}_2, \theta, M)$ on arvioitu oikein, bayesiläisen päätösteorian mukaan toimivan agentin toiminta on optimaalista, tulkittuna siten, että tietyllä ajanjaksolla saavutettavan hyödyn odotusarvo on tätä politiikkaa käyttäen maksimaalinen. Vaikka päätöksissä käytettävä todennäköisyysjakauma oliskin arvioitu väärin, bayesiläiseen päätösteoriaan perustuvan toiminnan voidaan osoittaa silti olevan ristiriidattomasti rationaalista siinä mielessä, että ei ole mahdollista konstruoida sellaista päätöksentekotilanteiden jonoa, jonka tuloksena bayesiläinen päätöksentekijä häviäisi varmasti (ns. "Dutch book"-argumentti, katso esim. [10, 9, 32]). Lisäksi voidaan osoittaa että bayesiläinen päätösteoria on *ainoa* tässä mielessä rationaalinen päätöksentekopolitiikka [102].

Päätösteoriaa on kritisoitu siitä että eri tilanteista seuraavien haittojen arviointi on usein kohtuuttoman vaikeaa tai jopa mahdotonta. Tämä on usein totta: esimerkissä 8 haittojen arvottaminen oli yksinkertaista, koska niitä arvioitiin pelkästään yhtiölle muodostuvien kustannusten kautta, mutta jos mukaan olisi otettu työntekijälle koituvien kärsimysten arviointi, tilanne olisi ollut paljon monimutkaisempi. Päätösteoriassa toteutuu kuitenkin bayesiläisen mallintamisen pääperiaate, jonka mukaan meidän on kirjattava täsmällisesti kaikki ongelmanratkaisussa käyttämämme oletukset. Jos käyttämämme järjestelmä ei toimi toivotulla tavalla, tiedämme että syy on joko siinä, että olemme arvioineet haittavaikutukset väärin, tai siinä että emme ole pystyneet estimoimaan käytettyjä ehdollisia todennäköisyyksiä riittävän tarkasti. Päätösteoria antaa meille kuitenkin teoreettisesti optimaalisen päämäärän, jota voimme approksimoida parhaan kykymme mukaan, ja jos järjestelmämme ei toimi halutulla tavalla, voimme yrittää analysoida mikä tekemistämme approksimaatioista ei pidä paikkaansa.

Tässä yhteydessä on korostettava että ihmisen rooli bayesiläisessä mallintamisessa on ensiarvoisen tärkeä: valittavissa olevista vaihtoehdoista seuraavien haittojen (tai hyötyjen) arviointi on tehtävä joka jää viime kädessä ihmisen tehtäväksi. Päätösteoria voi antaa toimintasuosituksen annettujen haittavaikutusten kannalta, mutta ihmisen tehtäväksi jää arvioida tehtyjen oletusten paikkansapitävyys ja siten toimintasuositusten järkevyys.Tästä syystä epätäsmällistä päättelyä suorittavia järjestelmiä onkin ryhdytty kutsumaan yhä useammin *päätöksentukijärjestelmiksi (decision support systems)* entisen termin *asiantuntijajärjestelmä (expert system)* sijaan.

Luku 2

Bayesiläisen lähestymistavan arviointia

Bayesiläisen mallintamisen edut: ristiriidaton päättely, ylioppimisen välttäminen, tilastollisen aineiston ja asiantuntijatietämyksen yhdistäminen, puuttuvan tiedon luonteva käsittely, teoreettinen kehikko hybridimalleille. Vastauksia bayesiläisen lähestymistavan kritiikkiin. Vertailua esimerkkeihin perustuvaan päättelyyn ja ei-parametrisiin malleihin. Yhteys informaatioteoreettisiin lähestymistapoihin: MDL, MML ja stokastinen kompleksisuus.

Luvussa 1 esitettyä bayesiläistä mallinnuskehikkoa kohtaan voidaan esittää kritiikkiä kohdistamalla se joko parametriseen mallintamislähestymistapaan sinänsä, tai nimenomaan bayesiläiseen mallintamiseen. Luvussa 2.3 käsittelemme lyhyesti esimerkkeihin perustuvaa päättelyä ja ei-parametrisia malliperheitä, jotka eivät kuitenkaan osoittaudu kovin erilaisiksi lähestymistavoiksi parametriseen mallintamiseen verrattuna. Sitä ennen kertaamme luvussa 2.1 tärkeimmät bayesiläisen lähestymistavan tarjoamat edut, ja arvioimme baysiläisyyttä kohtaan esitettyä kritiikkiä luvussa 2.2. Luvussa 2.4 käsitellään informaatioteoreettista lähestymistapaa mallintamiseen, ja osoitetaan sen läheinen yhteys bayesiläisyyteen.

2.1 Bayesiläinen mallintamisen etuja

Bayesiläinen mallinnus tarjoaa teoreettisesti elegantin, yhtenäisen lähestymistavan kaikkiin oppivien ja älykkäiden järjestelmien rakentamisessa esiintyviin ongelmiin. Mikäli käytettävä malliperhe on lisäksi probabilistinen, saavutetaan lähestymistavalla monia käytännön sovellusten kannalta merkittäviä etuja:

Ristiriidaton kalkyyli pätäsmällisen tiedon käsittelemiseksi.
Todennäköisyyslaskenta tarjoaa epätäsmällisen tiedon käsittelemiseksi matemaattisen kalkyylin, missä uskomusastetta tiettyyn väitteeseen kuvataan reaaliluvulla väliltä [0,1]. Syntyvä kalkyyli on *ristiriidaton l. konsistentti*, mikä tarkoittaa sitä että arvioidessamme jonkin suureen todenperäisyyden astetta, on todennäköisyyslaskennan tarjoama vastaus aina yksikäsitteinen. Mielivaltaisen lukuja väliltä [0,1] manipuloivan numeerisen järjestelmän ristiriidattomuus ei suinkaan ole itsestäänselvyys. Itse asiassa voidaan osoittaa että tiettyjen oletusten vallitessa todennäköisyyslaskenta on ainoa olemassaoleva konsistentti epätäsmällistä tietoa käsittelevä kalkyyli. Vaikka todistuksen perustana olevia oletuksia ei hyväksyttäisikään, osoittaa tämä teoreettinen tulos että konsistentin kalkyylin luominen on erittäin vaikeaa (enemmän aiheesta löytyy lähteestä [133]).

Todennäköisyyslaskennan käyttäminen poistaa siten yhden epätäsmällistä päättelyä suorittavien järjestelmien ongelmista: jos esimerkiksi rakennamme sumean sääntökannan ja käytämme jotakin mielivaltaisesti valittua kalkyyliä (esimerkiksi jotakin min-max-yhdistelysääntöjen variaatiota), ja järjestelmä ei toimi toivotulla tavalla, emme voi tietää onko meidän syytä vaihtaa käytettyä kalkyylia, vai järjestelmän parametreja (jäsenyysasteita). Bayesiläisessä lähestymistavassa kalkyyli on yksikäsitteinen, joten järjestelmän puutteiden korjaamiseksi riittää tarkastella käytetyn mallin parametreja.

Ylioppimista välttävä mallinvalintakriteeri.
Kuten luvussa 1.2.1 todettiin, bayesiläinen mallinvalintakriteeri sisältää "automaattisen Occamin partaveitsen" minkä ansiosta mallin struktuurin kompleksisuus voidaan valita siten että mallin tarkkuus myös opetusjoukon ulkopuolella (mallin yleistyskyky) on mahdollisimman hyvä. Bayesiläinen oppiminen pyrkii siis automaattisesti tasapainoon mallien kompleksisuuden ja niiden kuvausvoiman välillä. Vaikka bayesiläistä mallinvalintakriteeriä on luonnollisinta käyttää aidosti probabilististen malliperheiden, kuten esimerkiksi Bayes-verkkojen tapauksessa (katso luku 3.3), voidaan kriteeriä soveltaa myös muiden malliperheiden, esimerkiksi neuroverkkojen [88, 96, 12, 14, 112] ja päätöspuiden [110, 100] oppimisessa.

Tilastollisen aineiston ja asiantuntijatietämyksen yhdistäminen luonnollisella, teoreettisesti tyydyttävällä tavalla.
Luvussa 1.2.4 hahmoteltiin kuinka asiantuntijatietämys voidaan yhdistää tilastolliseen aineistoon priorijakaumia käyttäen. Toisaalta monissa probabilistisissa malliperheissä (kuten esimerkiksi luvussa 3 esitettyjen Bayes-verkkojen tapauksessa) voidaan sekä mallistruktuureille että malliparametreille antaa

selkeä semanttinen tulkinta, jolloin malleja voidaan konstruoida myös suoraan asiantuntijatietämystä käyttäen, ilman tilastollista oppimista (toisin kuin neuroverkkojen ja muiden ns. "black box"-mallien tapauksessa). Luvussa 3.3 sovellamme näitä periaatteita käytännössä, ja esitämme kuinka Bayes-verkkomalleja voidaan konstruoida käyttäen pelkästään tilastollista aineistoa, pelkästään asiantuntijatietämystä, tai yhdistämällä nämä kaksi tietolähdettä.

Parametrien oppimisen nopeus.
Kuten luvussa 1.3 todettiin, bayesiläisessä mallintamisessa voidaan optimaaliset malliparametrit löytää monessa tapauksessa suoraan ilman aikaavievää iteratiivista oppimisprosessia (vrt. neuroverkkojen parametrien oppiminen backpropagation-algoritmilla). Luvussa 3.3.2 annamme yksinkertaisen laskukaavan, jonka avulla Bayes-verkkomallin parametrit voidaan laskea suoraan annetusta opetusjoukosta $\mathcal{D}$.

Tarvittavan tilastollisen aineiston vähyys.
Bayesiläinen mallintaminen mahdollistaa asiantuntijatietämyksen hyväksikäytön mallien oppimisprosessissa. Jos asiantuntijatietämystä on saatavilla, on tarvittavan tilastollisen aineiston määrä hyvin pieni — itse asiassa mallit voidaan konstruoida suoraan asiantuntijatietämyksen perusteella, ilman tilastollista oppimista! Toisaalta, vaikka asiantuntijatietämystä ei olisikaan saatavilla, todellisilla aineistoilla suoritetut empiiriset kokeet [74] ovat osoittaneet että bayesiläisessä oppimisessa jo hyvin pieni esimerkkiotos (parhaissa tapauksissa vain muutamista esimerkeistä muodostuva joukko) riittää hyvän tuloksen saavuttamiseksi parametrien oppimisessa.

Monipuoliset sovellusmahdollisuudet yhteisjakauman mallintamisen ansiosta.
Kuten luvussa 1.3 näimme, bayesiläinen mallintaminen mahdollistaa hyvin monenlaisten ongelmien kuvaamisen yhdessä ristiriidattomassa teoreettisessa kehikossa. Koska probabilistiset mallit kuvaavat mallinnetussa maailmassa esiintyvien tilanteiden yhteisjakaumaa, voidaan samaa probabilistista mallia käyttää joustavasti hyvin erityyppisissä sovelluksissa.

Mahdollisuus sekoittaa jatkuvia ja diskreettejä muuttujia.
Probabilistisissa malleissa voidaan käyttää joko moniarvoisia diskreettejä muuttujia, jatkuva-arvoisia muuttujia, tai molempia yhtäaikaa. Tähän kysymykseen palataan luvussa 3.4.

Puuttuvan tiedon käsittelymekanismit.
Kuten luvussa 1.3 näimme, probabilistinen lähestymistapa tarjoaa selkeän teoreettisen kehikon puuttuvan tiedon käsittelemiseksi: jos puuttuva tieto ei sinänsä ole kiinnostuksen kohteena, probabilistinen päättely voidaan suorittaa olemassaolevan tiedon perusteella marginalisoimalla (integroimalla) puuttuvan tiedon vaikutus pois. Puuttuva tieto voidaan toisaalta myös täydentää optimaalisella tavalla vertailemalla eri täydennysten todennäköisyyksiä.

Toimintavaihtoehtojen päätösteoreettinen analyysi.
Vaihtoehtoisista lähestymistavoista poiketen bayesiläinen päätösteoria tarjoaa mahdollisuuden arvioida eri toimintavaihtoehtoihin liittyviä odotettavia riskejä ja hyötyjä. Päätösteoriaa voidaan käyttää odotusarvoisesti pienimmän riskin (tai suurimman hyödyn) tuottavan vaihtoehdon etsimiseksi luvussa 1.4 esitetyllä tavalla, tai jo tehdyn päätöksen seurauksien analysoimiseksi. Päätösteoriaa voidaan luonnollisesti soveltaa myös ei-probabilististen mallien antamien tulosten käsittelyssä, jos mallien tulokset esitetään todennäköisyysjakaumana. On kuitenkin huomattava että pelkkä numeroiden suora normeeraus siten että niiden summa on yksi ei välttämättä tuota todennäköisyysjakaumaa joka vastaisi ongelmakentän todellista jakaumaa hyvin, eikä päätösteorian soveltaminen tällaiselle jakaumalle tuota hyviä tuloksia.

Teoreettinen kehikko hybridimallien konstruoimiseksi.
Luvussa 1.2 esitetyssä lähestymistavassa valittiin annetun opetusdatan $\mathcal{D}$ perusteella yksi mallistruktuuri M, ja yksi parametriarvoilmentymä $\hat{\theta}$, joiden avulla konstruoitiin ongelmakenttää mallintava yhteistodennäköisyysjakauma $P(X_1 = x_1, \ldots, X_n = x_n \mid M, \hat{\theta}, \mathcal{D})$. On kuitenkin huomattava, että vaikka esitetty bayesiläinen kriteeri valitsee kaikkein todennäköisimmän parametrivektorin $\hat{\theta}$, saattaa olla mahdolllista että on muita parametriarvovektoreita, jotka ovat lähes yhtä todennäköisiä kuin $\hat{\theta}$. Jos päämääränä on tehdä tarkkoja ennustuksia saatavilla olevan aineiston $\mathcal{D}$ perustella, on helppo nähdä että ennustusten tarkkuutta voidaan parantaa käyttämällä yhden mallin sijasta joukkoa todennäköisiä parametrivektoreita, painottaen niiden antamia tuloksia parametrivektorien todennäköisyydellä. Itse asiassa bayesiläinen lähestymistapa ei pysähdy edes vielä tähän, vaan tarkasti ottaen bayesiläisessä mallintamisessa edellytetään, että optimaalisen ennustustarkkuuden saavuttamiseksi on käytettävä jakaumaa ,joka saadaan laskemalla integraali

("painotettu summa") yli *kaikkien* (äärettömän monen) parametrivektorin θ:

$$P(X_1 = x_1, \ldots, X_n = x_n \mid M, \mathcal{D}) \\ = \int P(X_1 = x_1, \ldots, X_n = x_n \mid M, \theta) P(\theta \mid M, \mathcal{D}) d\theta.$$

Tämä lähestymistapa johtaa odotusarvoparametrien käyttöön, joista puhutaan luvussa 3.3.2. Toisaalta vastaavaa ajatuskulkua voidaan soveltaa myös mallistruktuurin valinnassa, jolloin päädytään käyttämään jakaumaa

$$P(X_1 = x_1, \ldots, X_n = x_n \mid \mathcal{D}) \\ = \sum_M (X_1 = x_1, \ldots, X_n = x_n \mid M, \mathcal{D}) P(M \mid \mathcal{D}).$$

Bayesiläisen filosofian mukaan opetusdatan perusteella saadaan tarkin ongelmakentän jakauman esitys muodostamalla painotettu summa saatavilla olevista mallistruktuureista, kun painokertoimina käytetään mallistruktuurien posterioritodennäköisyyksiä $P(M \mid \mathcal{D})$. Näin voidaan yhdistää esimerkiksi monen eri Bayes-verkon antamat tulokset, tai muodostaa vaikkapa Bayes-verkon, neuroverkon, ja päätöspuun muodostama hybridimalli. Jälkimmäisessä tapauksessa ongelmaksi saattaa kuitenkin muodostua se, kuinka konstruoida jakaumamalli, joka antaa eri malliperheistä tuleville mallistrktuureille yhteismitalliset todennäköisyydet. Lisäksi on huomattava että hybridimallien tulkittavuus ei ole kovin hyvä, joten lähestymistapaa ei voida käyttää ongelmakentän analysointiin tähtäävissä sovelluksissa (data mining).

2.2 Bayesiläinen lähestymistavan kritiikkiä

Edellisessä luvussa esitetyistä bayesiläisen mallintamisen eduista huolimatta on bayesiläinen mallinnus toistaiseksi jäänyt ehkä vähemmälle huomiolle kuin vaihtoehtoiset lähestysmistavat. Yksi syy on se, että bayesiläiseen mallinnukseen liittyy sen formaalin luonteen vuoksi joitakin teknisiä ongelmia, mutta erityisesti luvussa 3 kuvatun Bayes-verkkomalliperheen syntyminen on poistanut monet näistä ongelmista. Saavutetut tulokset ovat kuitenkin verrattain uusia (Bayes-verkkoteoriaa on tutkittu laajemmin vasta noin 10 vuoden ajan), eivätkä siksi vielä kovin laajalti tunnettuja. Maailmalla liikkuukin bayesiläisestä mallintamisesta monia harhakäsityksiä, jotka eivät pidä paikkaansa.

Seuraavassa kokoelma yleisimpiä bayesiläiseen lähestymistapaan kohdistuvia kriittisiä kommentteja.

Bayesiläinen mallinnus edellyttää epärealistisen suuren parametrijoukon arvojen määräämistä.
Bayesiläinen mallinnuksen myönnetään usein olevan *teoriassa* hieno lähestymistapa, mutta probabilististen mallien uskotaan edellyttävän niin suuren parametrijoukon arvojen tarkkaa määrittämistä, että lähestymistapa ei ole käytännössä mahdollinen. Luvussa 3 esitetyt Bayes-verkkomallit ovat vasta muutamia vuosia sitten kehitetty elegantti ratkaisu tähän ongelmaan: Bayes-verkkoja käyttäen tarvittavien parametrien määrä saadaan vähennettyä käytännön sovellukset mahdollistavalle tasolle. Toisaalta, kuten edellä jo todettiin, bayesiläinen mallintaminen tarjoaa mahdollisuuden oppia malleja suoraan tilastollisesta aineistosta, suoraan asiantuntijatietämystä soveltaen, tai yhdistämällä molemmat tietolähteet. Tämä mahdollisuus siirtää parametrien määräämistehtävän ihmiseltä koneelle suoritettavaksi. Lisäksi on vielä todettava että bayesiläisten mallien on empiirisesti havaittu kestävän hyvin parametrien arvojen epätarkkuutta, jolloin parametrien tarkka määrittäminen ei ole tärkeää—käytännön sovelluksia ajatellen on usein riittävää kunhan ne ovat oikeaa suuruusluokkaa [57].

Priorijakaumien määrääminen on vaikeaa.
Kuten edellä näimme, bayesiläinen mallintaminen edellyttää että järjestelmälle annetaan kaksi priorijakaumaa: mallistruktuurien priorijakauma $P(M)$ ja malliluokan parametrien priorijakauma $P(\theta \mid M)$. Tämä seikka mainitaan usein bayesiläisen mallintamisen heikkoutena, vaikka tosiasiassa se on yksi lähestymistavan suurimpia etuja — mahdollistaahan priorijakaumien käyttö nimenomaan tilastollisen tiedon ja asiantuntijatietämyksen yhdistämisen luonnollisella tavalla. Mikäli prioritietämystä ei ole saatavilla, hyvä vaihtoehto käytännössä on käyttää ns. ei-informatiivista, tasaista priorijakaumaa. Toinen käyttökelpoinen vaihtoehto on käyttää informaatioteoreettisia lähestymistapoja joita tarkastellaan luvussa 2.4.

Bayesiläinen päättely on epäintuitiivista.
Bayesiläistä lähestymistapaa on kritisoitu siitä, että sen tarjoama epätäsmällisen päättely olisi jollakin tavoin "epäintuitiivista", erilaista kuin ihmisten suorittama epätäsmällinen päättely. Suurin syy tähän uskomukseen on vanha objektivistinen todennäköisyyskäsitteen tulkinta, jonka mukaan todennäköisyydet määritellään toistokokeissa esiintyvinä frekvensseinä. Uudemman, *subjektivistisen* todennäköisyystulkinnan mukaan todennäköisyydet määritellään uskomusastetta kuvaavina lukuina väliltä $[0, 1]$. Tämän ansiosta voidaan subjektivistisessa lähestymistavassa puhua esimerkiksi todennäköisyydestä sille että maailmanloppu tulee vuonna 2000, kun taas vanhassa fre-

kventistisessä mallissa kyseisen todennäköisyyden käyttäminen oli ongelmallista. Luvussa 3.2 annamme probabilistisesta päättelystä esimerkkejä, jotka tukevat intuitiivisesti hyvin luonnolliselta tuntuvia päättelyn muotoja (esimerkiksi ns. syiden poisselittämisilmiötä).

Myös lukuisat päätösteoriaan liittyvät paradoksit aiheuttavat sekaannusta. tyypillisessä esimerkissä koehenkilölle annetaan kaksi vaihtoehtoa: vaihtoehdossa A hän saa kymmenen miljoona markkaa todennäköisyydellä 1 (siis varmasti), kun taas vaihtoehdossa B hän saa kymmenen miljoona markkaa todennäköisyydellä 0.25, neljäkymmentä miljoonaa markkaa todennäköisyydellä 0.25, ja ei mitään todennäköisyydellä 0.5. Vaihtoehto A on varmaankin itse kunkin mielestä paljon houkuttelevampi, vaikka päätösteoria suosittelee vaihtoehdon B valitsemista, koska voiton odotusarvo on yli kymmenen miljoona markkaa! Esimerkki paljastaa kuinka tärkeää päätösteoriassa on odotettavissa olevan hyödyn mittaaminen oikein: siinä oletettavassa lähtötilanteessa, jossa koehenkilö ei ole miljonääri, ei neljänkymmenen miljoonan markan voittaminen ole hänelle neljä kertaa arvokkaampaa kuin kymmenen miljoonan voittaminen: jo kymmenen miljoonaa riittää siihen että henkilö voi elää loppuelämänsä mukavissa olosuhteissa ilman tarvetta palkkatyöhön, joten voittosumman kasvattaminen nelinkertaiseksi ei tuo merkittävää lisäystä voiton aikaansaamaan elämänmuutokseen. Jos eri markkamääriä vastaavat todelliset arvostukset otetaan huomioon, päätösteoria toimii inhimillisesti katsoen "järkevällä" tavalla.

Toisaalta on vielä todettava, että vaikka bayesiläinen päättely olisikin jossakin mielessä epäintuitiivista, ehkäpä se olisi vain hyvä asia: ihmiset ovat monesti osoittautuneet huonoiksi päätöksentekijöiksi monimutkaisia suureita käsittelevissä tilanteissa. Kuten edellä kuvattu "paradoksi" osoittaa, ihmiselle jää oppivien ja älykkäiden järjestelmien käyttämisessä tärkeä rooli: on vaikeaa kuvitella tietokoneohjelmaa joka pystyisi tyydyttävästi määräämään ongelmakentästä tärkeimmät olemassaolevat toimintavaihtoehdot, ja arvioimaan niihin liihin liittyviä hyötyjä ja haittoja.

Bayesiläinen mallintaminen käsittelee vain tiedon epävarmuutta, ei tiedon epätäsmällisyyttä.

Todennäköisyyslaskentaa on kritisoitu siitä, että se käsittelee perinteisiä, täsmällisiä objekteja, eikä epätäsmällisyyttä kuvaavia ns. sumeita käsitteitä (katso esim. [58]). Koska todennäköisyysjakaumia voidaan määritellä mielivaltaisen moniarvoisille tai jatkuva-arvoisille muuttujille, voidaan bayesiläisessä mallintamisessa kuitenkin käsitellä samoja kysymyksiä kuin sumeassa logiikassakin [21, 20].

Bayesiläisen mallintamisen soveltaminen on vaikeaa.
Bayesiläisen mallintamisen edellyttämä kaikkien oletusten eksplisiittisen formalisoinnin täydellinen ymmärtäminen edellyttää tiettyjä matemaattisia valmiuksia. Tällaisen mallintamisen soveltaminen käytännössä ei kuitenkaan edellytä näiden teknisten yksityiskohtien syvempää ymmärrystä, vaan bayesiläisen mallintamisen teoriaa voidaan käyttää tuottamaan yksinkertaisia laskenta-algoritmeja, kuten lukuisat saatavilla olevat ohjelmistot todistavat (ks. luku 4.2). Toisaalta monet heuristisesti kehitetyt "ad hoc"-menetelmät voidaan tulkita todennäköisyyslaskennan tarjoamassa kehikossa. Tämän avulla on pystytty esimerkiksi selittämään ne epärealistiset riippumattomuusoletukset, jotka joudutaan (implisiittisesti) tekemään käytettäessä ns. varmuuskertoimiin perustuvia sääntökantoja [46].

2.3 Esimerkkeihin perustuva päättely ja ei-parametriset mallit

Ei-parametrisissa (non-parametric) lähestymistavoissa luovutaan mallien eksplisiittisestä, mallistruktuurin kiinnittämästä parametroidusta esitysmuodosta, ja yritetään ratkaista ongelmakentässä esiintyvät tehtävät suoraan esimerkkitapauksia soveltaen. *Esimerkkeihin perustuvassa päättelyssä (case-based reasoning, CBR)*[1] päättelyongelmat ratkaistaan kaksivaiheisella prosessilla. Ensimmäisessä, *esimerkkien vertailuvaiheessa (case matching)*, käsillä olevaa tilannetta verrataan muistissa oleviin aikaisempiin esimerkkitapauksiin (cases), ja tapauksille annetaan yhteensopivuusaste (matching score), joka on sitä suurempi mitä enemmän muistissa oleva tapaus muistuttaa nykyistä tilannetta. Päättelyn toisessa, *esimerkkien sovellusvaiheessa (case adaptation)* muodostetaan ratkaisuehdotus muokkaamalla suurimman yhteensopivuusasteen saaneiden vanhojen tapausten ratkaisemisessa käytetyistä menetelmistä ehdotus nykyisen tilanteen ratkaisemiseksi.

Yksinkertaisin esimerkki CBR-tyyppisestä lähestymistavasta on ns. *lähimmän naapurin menetelmä (nearest neighbor method)*, jossa esimerkkien yhteensopivuusasteena käytetään niiden (euklidista) etäisyyttä, ja tarvittavat tiedot kopioidaan suoraan korkeimman yhteensopivuusasteen saaneesta esimerkkitapauksesta. Monimutkaisemmissa malleissa tarvittavat tiedot muodostetaan käyttämällä opetusjoukon tapauksien painotettua summaa, käyttäen painokertoimina tapausten yhteensopivuusasteita. Esimerkkeihin

[1] Lisätietoa esimerkkeihin perustuvasta päättelystä löytyy esimerkiksi lähteistä [68, 139]. Lähestymistavasta käytetään myös termejä *memory-based reasoning* [131],*instance-based learning* [2], *lazy learning* [1] ja *transductive inference* [137].

perustuvan päättelyn toimintaperiaate on hyvin samanlainen kuin tilastotieteessa *ydinestimaattoreiksi (kernel estimators)* kutsutuilla menetelmillä (katso esim. [121]). Ydinestimaattoreissa muodostetaan annetun opetusjoukon kunkin tapauksen ("ytimen") ympärille jakauma jonka keskikohta on ytimessä, ja kokonaisjakauma saadaan tällaisten ydinjakaumien summana. Neuroverkkotutkimuksen alueella vastaavia menetelmiä kutsutaan *ydinkantafunktioiksi (radial basis functions)* [92, 107] tai *probabilistisiksi neuroverkoiksi (probabilistic neural networks)* [128].

On tärkeää huomata että nimestään huolimatta ei-parametriset mallit eivät suinkaan ole parametrittomia: esimerkiksi vaikka ydinestimaattoreissa on ydinjakaumien määrä, ja niiden keskikohta kiinnitetty, on jakaumien tarkka muoto määrättävä joukolla parametreja (esim. ydinjakaumakohtaiset varianssit), jotka on arvioitava esimerkkiaineistosta. Esimerkkeihin perustuvassa päättelyssä puolestaan on tapausten yhteensopivuusasteen laskemisessa käytettävä metriikka käyttäjän vapaasti määrättävissä, ja eri metriikat voidaan ajatella jonkin (mahdollisesti hyvin monimutkaisen) malliperheen parametrisoituina malleina. Se, halutaanko puhua eri metriikoista ja mallien soveltamisessa käytettävissä laskentamenetelmistä, vai parametrisista malleista, näyttää olevan pitkälti makuasia.

Bayesiläisiin parametrisiin malleihin perustuvilla lähestymistavoilla on se etu, että menetelmä pakottaa määrittelemään täsmällisesti kaikki oletukset, joiden varassa rakennettavan järjestelmän toiminta on. Ilman tällaista lähestymistapaa päädytään ad hoc-tyyppisiin menetelmiin, joiden toimivuudesta ei ole teoreettisia takeita. Voidaan esimerkiksi kysyä mihin perustuu useimmissa CBR-menetelmissä implisiittisesti tehtävä oletus, jonka mukaan euklidisen etäisyysmitan mielessä toisiaan muistuttavien tapausten ratkaisemisessa voidaan soveltaa samoja menetelmiä? Tämä oletus saattaa tietenkin pitää paikkansa ongelmasta, ja sen tapausten koodaamisessa käytetystä menetelmästä riippuen, mutta menetelmien käyttäminen ilman tehtyjen implisiittisten oletusten tiedostamista saattaa johtaa vakaviin virhepäätelmiin.

Ei-parametristen lähestymistapojen ero parametriseen mallintamiseen näyttää olevan siis lähinnä siinä, että ei-parametrisissa malleissa parametrien lukumäärä ei ole kiinteä, vaan se riippuu käytettävissä olevan opetusjoukon koosta. Kuten edellä kuitenkin näimme, baysiläisessä mallintamisessa on mallistruktuurin valinta kiinteä osa oppimisprosessia, ja valittava mallistruktuuri (ja sitä kautta malliparametrien lukumäärä) riippuu käytettävästä opetusjoukosta[2]! Parametristen ja erilaisten ei-parametristen lähestymistapojen ero ei siis näytä olevan periaattellinen, vaan lähinnä terminologinen. Itse asiassa

[2]Tästä syystä tähän periaatteeseen perustuvia malleja kutsutaan joskus *semiparametrisiksi (semi-parametric)* malleiksi.

sekä esimerkkeihin perustuva päättely että ydinestimaatorit (ja muut vastaavat mallit) voidaan ajatella erikoistapauksena *äärellisistä sekajakaumista (finite mixture models)* [35, 136], joita käsitellään enemmän luvussa 3.4. Äärellisten sekajakaumamallien ja esimerkkeihin perustuvan päättelyn samankaltaisuus osoitetaan lähteissä [135, 134, 95]. Yleisempää todennäköisyyslaskentaan perustuvaa formalismia esimerkkeihin perustuvalle päättelylle on ehdotettu lähteessä [69]. On huomattava, että ehdotetut bayesiläiset formalismit eivät välttämättä noudata esimerkkeihin perustuvan päättelyn normaalia kaksivaiheista päättelyprosessia, vaan päättely voi tapahtua suoraan ilman esimerkkien yhteensopivuusasteen laskemista. Bayesiläinen menetelmä tapausten yhteensopivuusasteen laskemiseksi esitetään lähteessä [70].

2.4 Informaatioteoreettiset lähestymistavat: MDL ja MML

2.4.1 MDL-periaate ja stokastinen kompleksisuus

Mallistruktuurin valinnassa voidaan käyttää myös informaatioteoreettista lähestymistapaa, joka on hyvin läheisessä suhteessä bayesiläiseen mallintamiseen. Rissasen *Minimum Description Length (MDL)*-periaatteen [110] mukaan mallistruktuureista valitaan se, jonka avulla opetusjoukko $\mathcal{D}$ voidaan koodata siten että syntyvä koodi on mahdollisimman lyhyt. Rissanen kutsuu lyhintä tällaista koodinpituutta opetusjoukon $\mathcal{D}$ *stokastiseksi kompleksisuudeksi (stochastic complexity)*, annettuna mallistruktuuri M. Tällä koodausteoreettiselle lähestymistavalle voidaan antaa intuitiivisesti selkeä tulkinta: lyhimmän koodin tuottava mallistruktuuri on ongelmakentän paras kuvaus siksi että lyhyen koodin aikaansaamiseksi on mallistruktuuriin koodattava kaikki mahdolliset aineistossa $\mathcal{D}$ esiintyvät säännönmukaisuudet.

Kuinka stokastinen kompleksisuus sitten määritellään? Ensiksikin on korostettava, että on mahdotonta konstruoida määritelmää joka tuottaisi lyhimmän koodin kaikille mahdollisille opetusjoukoille $\mathcal{D}$: koodausteorista tiedämme että jos $L(\mathcal{D} \mid M)$ kuvaa opetusjoukkoa $\mathcal{D}$ vastaavan koodin pituutta mallistruktuurin M tapauksessa, määrittelee

$$P(\mathcal{D} \mid M) = 2^{-\log L(\mathcal{D} \mid M)}$$

todennäköisyysjakauman yli kaikkien mahdollisten joukkojen $\mathcal{D}$. Koska

$$\sum_{\mathcal{D}} P(\mathcal{D}|M) = 1,$$

ei ole mahdollista että olisi todennäköisyysjakauma P^* jolle pätisi: $P^*(\mathcal{D}|M) > P(\mathcal{D}|M)\ \forall P, \mathcal{D}$. Siispä ei myöskään voi olla sellaista koodia L, joka tuottaisi lyhimmän mahdollisen kuvauksen kaikille joukoille $\mathcal{D}$. Stokastinen kompleksisuus täytyykin määritellä siten että se tuottaa lyhyimmän kuvauksen 'tyypillisimmille' joukoille $\mathcal{D}$. Se kuinka 'tyypillisyys' määritellään, ratkaisee viime kädessä eri stokastisten määritelmien hyvyyden.

Varhaisessa tuotannossaan [109] Rissanen määritteli stokastisen kompleksisuuden seuraavasti:

$$L(\mathcal{D} \mid M) = -\log P(\mathcal{D} \mid M, \tilde{\theta}) + \frac{d(M)}{2} \log N, \tag{2.1}$$

missä $(M, \tilde{\theta})$ on malliluokan M *suurimman uskottavuuden (maximum likelihood)*-malli, ts. malli, joka maksimoi uskottavuustermin $P(\mathcal{D} \mid M, \theta)$, $d(M)$ on mallin parametrien lukumäärä, ja N opetusjoukon $\mathcal{D}$ koko. Näemme että tämä informaatioteoreettinen stokastisen kompleksisuuden määritelmä antaa saman tuloksen kuin luvussa 1.2.2 esitetty kokonaisuskottavuuden BIC-approksimaatio.

Myöhemmin(katso esim. [110]) Rissanen osoitti, että määritelmä

$$L(\mathcal{D} \mid M) = -\log P(\mathcal{D} \mid M) = -\log \int P(\mathcal{D} \mid M, \theta) P(\theta \mid M) d\theta \tag{2.2}$$

tuottaa koodeja, joiden voidaan osoittaa olevan monessa suhteessa parempia kuin määritelmän (2.1) tuottamat koodit. Määritelmästä (1.1) näemme että tämä kriteeri vastaa bayesiläistä lähestymistapaa, jossa mallistruktuurien prioriksi on oletettu tasainen jakauma, ts. $P(M_1) = P(M_2) = \ldots = P(M_K)$, ja mallistruktuurin valintakriteerinä käytetään siis yksinomaan kokonaisuskottavuutta $P(\mathcal{D} \mid M)$. Äskettäin [111] Rissanen esitteli uuden, tietyssä mielessä vielä tehokkaamman koodin, jossa stokastinen kompleksisuus määritellään seuraavasti:

$$L(\mathcal{D} \mid M) = -\log \frac{P(\mathcal{D} \mid \tilde{\theta}(\mathcal{D}), M)}{\sum_{\mathcal{D}'} P(\mathcal{D}' \mid \tilde{\theta}(\mathcal{D}'), M)}, \tag{2.3}$$

missä $\tilde{\theta}(\mathcal{D})$ on opetusjoukkoa $\mathcal{D}$ vastaava suurimman uskottavuuden malli, ja summa nimittäjässä käy yli kaikkien mahdollisten opetusjoukkojen. Voidaan osoittaa että tämä uusi stokastisen kompleksisuuden määritelmä antaa asymptoottisesti saman tuloksen kuin määritelmä (2.2), jos käytämme integraalissa (2.2) parametrien priorijakaumana ns. Jeffreyn prioria. Tätä kysymystä käsitellään tarkemmin lähteissä [75, 76].

2.4.2 MML-periaate

Wallacen kehittämä *MML (minimum message length)*-periaate on läheistä sukua Rissasen MDL-periaatteelle, mutta MML-lähestymistavassa keskitytään malliparametrien valitsemisen ongelmaan, kun taas MDL-periaatetta sovelletaan yleensä mallistruktuurin valintaongelmaan. MML-lähestymistavassa hylätään edellä esitetty mallien kaksivaiheinen oppiminen, ja sulautetaan kaikki malliluokat M yhdeksi suureksi kaikkien mallien muodostamaksi malliperheeksi. Koska malliluokan käsitettä ei tässä lähestymistavassa käytetä, malliperhe muodostuu erikokoisista malleista eli eripituisista parametrivektoreista θ.

MML-periaatteen mukaan *paras malli $\bar{\theta}$ on se joka tuottaa lyhimmän koodin kun koodataan käytetty malli $\bar{\theta}$, ja opetusaineisto $\mathcal{D}$ käyttäen mallia $\bar{\theta}$*:

$$\bar{\theta} = \arg\min_{\theta} L(\theta) + L(\mathcal{D} \mid \theta).$$

Jos mallien parametrit oletetaan reaaliluvuiksi, ei koodia kuitenkaan voida konstruoida, koska puhtaiden reaalilukujen esittäminen edellyttää kykyä esittää lukuja äärettömän tarkasti, mikä taas vaatii äärettömän pitkiä koodeja. Siksi MML-lähestymistavassa oletetaan että mallien parametrit pystytään esittämään vain äärellisellä tarkkuudella (mikä tietysti on realistinen oletus käytännön sovelluksia ajatellen), ja mallit θ diskretisoidaan äärellisen koodin saamiseksi. Diskretisointi on tehtävä niin, että syntyvän koodin pituus on mahdollisimman lyhyt.

Baysiläisessä oppimisessa toisaalta yksittäisen mallin θ soveltuvuus mitattiin sen posterioritodennäköisyydellä

$$P(\theta \mid \mathcal{D}) = \frac{P(\mathcal{D} \mid \theta)P(\theta)}{P(\mathcal{D})}. \tag{2.4}$$

Jos unohdamme nimittäjässä olevan vakion $P(\mathcal{D})$, saamme

$$-\log P(\theta \mid \mathcal{D}) \propto -\log P(\mathcal{D} \mid \theta) - \log P(\theta).$$

Kun merkitsemme $L(\theta) = -\log P(\theta)$ ja $L(\mathcal{D} \mid \theta) = -\log P(\mathcal{D} \mid \theta)$, näemme että MML-periaate on bayesiläinen siinä mielessä että se valitsee mallin joka maksimoi posterioritodennäköisyyden (2.4), kun mallit on diskretisoitu optimaalisella tavalla. MML-periaatteesta löytyy lisämateriaalia lähteistä [99, 98, 6], ja Monash-yliopiston tutkimusryhmän ylläpitämältä WWW-sivulta[3].

[3]URL: http://www.cs.monash.edu.au/~lloyd/tildeMML/'

Luku 3

Bayes-verkot

Mitä ovat Bayes-verkot? Kausaalinen mallintaminen Bayes-verkoilla. Päättely epätäydellisillä tiedoilla Bayes-verkkoja käyttäen. Bayes-verkkojen muodostaminen automaattisesti: verkon rakenteen oppiminen ja parametrien oppiminen. Bayes-verkkojen muunnelmia: päätösverkot, jatkuva-arvoiset Bayes-verkot, Naiivi Bayes-malli, kvalitatiiviset Bayes-verkot.

Kuten kuvasta 1.1 näemme, ensimmäinen askel bayesiläisessä mallinnusprosessissa on malliperheen valinta. Vaikka bayesiläistä mallinnusta voi periaatteessa soveltaa myös ei-probabilistisiin malliperheisiin, kuten esimerkiksi neuroverkkoihin (ks. [88, 96, 12, 14, 113]), saavutetaan bayesiläisen lähestymistavan edut helpoimmin käyttäen probabilistisia malleja. *Bayes-verkot* (Bayesian networks, Bayesian belief networks) muodostavat tällaisen malliperheen, joka on saavuttanut viime aikoina suurta huomiota. Pääsyy Bayes-verkkoihin kohdistuvassa kiinnostuksessa johtuu 80-luvulla kehitetystä teoreettisesta kehikosta, jonka avulla probabilististen mallien luomiseen tarvittavien parametrien määrää voidaan huomattavasti pienentää. Luvussa 3.1 selvitetään lyhyesti Bayes-verkkojen pääperiatteet, puuttumatta tässä yhteydessä tarkemmin teknisiin yksityiskohtiin, jotka selviävät esimerkiksi lähteistä [105, 97, 60, 18]. Luvussa 3.2 näemme kuinka luvussa 1.3 esitettyjä probabilistisen päättelyn eri muotoja on mahdollista toteuttaa Bayes-verkkomallien avulla. Bayes-verkkojen konstruoimista käsitellään luvussa 3.3.

3.1 Bayes-verkkomalliperhe

Esitämme seuraavassa Bayes-verkkomalliperheen perusominaisuudet yksinkertaistettujen esimerkkien avulla. Tässä yhteydessä on korostettava, että vaikka käytetyissä esimerkeissä esiintyy yleensä vain binäärisiä diskreettejä muuttujia, voidaan lähestymistavassa käyttää myös sekä moniarvoisia diskreettejä muuttujia että jatkuvia muuttujia. Tähän kysymykseen palaamme luvussa 3.4.

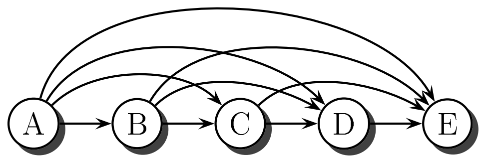
Oletetaan seuraavassa esimerkissä että ongelmakentän kuvauksessa käytetään viittä binääristä attribuuttia (satunnaismuuttujaa) A, B, C, D, E, ja käytetään binäärimuuttujien mahdollisista arvoista merkintöjä 0 ja 1. Tässä matemaattisessa mallissa maailman kaikki mahdolliset tilat koostuvat muuttujien A, B, C, D, E arvojen muodostamista binäärivektoreista, joita on tässä yksinkertaisessa viiden binäärimuuttujan tapauksessa kaikkiaan $2^5 = 32$ kappaletta. Bayesiläisessä mallinnuksessa on päämääränä muodostaa matemaattinen malli, jonka avulla voidaan määrätä todennäköisyys mille tahansa mahdolliselle maailman tilalle (5-komponenttiselle binäärivektorille). Tällaista *yhteistodennäköisyysjakaumaa* hyväksikäyttäen voidaan suorittaa luvussa 1.3 esitettyjä probabilistisen päättelyn eri muotoja, kuten luvussa 3.2 tulemme näkemään.

Bayesiläisen mallintamisen päämääränä oleva yhteistodennäköisyysjakauma voidaan luonnollisesti määrätä luettelemalla kaikki mahdolliset tilat ja niitä vastaavat todennäköisyydet, mutta koska tilojen lukumäärä on eksponentiaalinen suhteessa muuttujien lukumäärään, ei tämä lähestymistapa ole käytännössä esiintyvissä tilanteissa mahdollinen. Toisaalta todennäköisyyslaskennan perusaksioomia käyttäen on helppo osoittaa että

$$P(A,B,C,D,E) = \\ P(A)P(B|A)P(C|A,B)P(D|A,B,C)P(E|A,B,C,D). \quad (3.1)$$

Muuttujien A, B, C, D, E yhteistodennäköisyysjakauman määräämiseksi siis riittää, että tunnemme yhtälössä (3.1) oikealla puolella esiintyvät ehdolliset todennäköisyydet. Tässä yhteydessä on huomattava, että se mitä ehdollisia todennäköisyyksiä tarvitaan riippuu käytetystä muuttujien järjestyksestä. Jos yhtälöä (3.1) sovelletaan esimerkiksi muuttujien järjestykseen (E, D, C, B, A), saadaan yhtälön oikealle puolelle ehdolliset todennäköisyydet $P(E)$, $P(D \mid E)$, $P(C \mid E, D)$, $P(B \mid E, D, C)$, ja $P(A \mid E, D, C, B)$. Käytettyä permutaatiota (ja siis sitä vastaavaa ehdollisten todennäköisyyksien joukkoa) voidaan havainnollistaa graafisesti: kuvassa 3.1 on esitetty syklitön suunnattu verkko, jossa muuttujasta Y on muuttujaan X kaari ainostaan silloin kun jokin hajotelmassa (3.1) käytettävistä ehdollisista todennäköisyyksistä on muotoa $P(X \mid *, Y, *)$.

Kuvassa 3.1 esitetty suunnattu verkko määrää siis minkä tyyppisiä ehdollisia todennäköisyyksiä on käytettävä, jos halutaan määrätä muuttujien yhteisjakauma $P(A, B, C, D, E)$ käyttäen ketjusääntöä (3.1). Kiinnittämäl-



Kuva 3.1: Hajotelman (3.1) esitys suunnattuna verkkona.

lä verkkoesitystä vastaavien ehdollisten todennäköisyyksien arvot on yhteisjakauma yksikäsitteisesti määrätty. Toisaalta, kuten luvussa 1.2 esitettiin, bayesiläisessä mallintamisessa malli määritellään parina (M, θ), missä M eli mallin struktuuri määrää mitä parametreja mallin identifioimiseksi on määrättävä, ja θ on joukko parametriarvoja jotka kiinnittävät yhden mallin. Kuvassa 3.1 esitettyä graafista verkkoesitystä voidaan siis ajatella mallin struktuurina, ja graafia vastaavia ehdollisia todennäköisyyksiä voidaan pitää mallin parametreina.

Ottaen huomioon että $P(X = 1 \mid *) = 1 - P(X = 0 \mid *)$ voimma laskea että käyttäen kuvassa 3.1 esitettyä mallistruktuuria tarvitsemme $1 + 2 + 4 + 8 + 16 = 31$ parametria (ehdollisen todennäköisyyden arvoa) yhteistodennäköisyysjakauman määräämiseksi. Koko avaruuden tilojen lukumäärään (32) nähden emme ole siis säästäneet paljoa. Keskeinen tekijä Bayes-verkkoteoriassa onkin ehdollisen riippumattomuuden käsite, jota käyttäen tarvittaen parametrien määrää voidaan radikaalisti pienentää:

Määritelmä 1 (Ehdollinen riippumattomuus)
Olkoot $\mathbf{X}$, $\mathbf{Y}$, *ja* $\mathbf{Z}$ *erillisiä muuttujajoukkoja. Joukon* $\mathbf{X}$ *sanotaan olevan* ehdollisesti riippumaton *muuttujajoukosta* $\mathbf{Y}$, *annettuna joukko* $\mathbf{Z}$, *jos*

$$P(\mathbf{X} \mid \mathbf{Y}, \mathbf{Z}) = P(\mathbf{X} \mid \mathbf{Z}).$$

Selventääksemme ehdollisen riippumattomuuden käsitettä olettakaamme esimerksi että ongelmakentäämme liittyy lääketieteelliseen diagnosointiin, ja esimerkkimuuttujillamme A, B, C, D, E on taulukossa 3.1 esitetyt semanttiset tulkinnat[1].

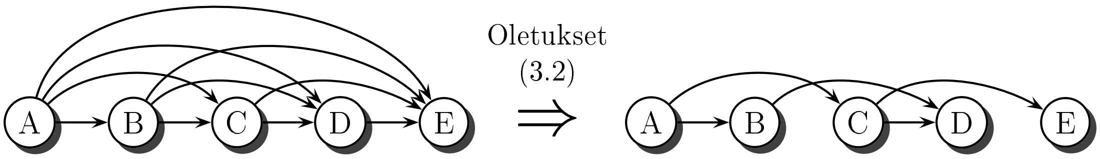
Ongelmakentän hyvin tunteva asiantuntija saattaisi kertoa muuttujien suhteesta esimerkiksi seuraavaa:

> Vakavien päänsärkykohtausten todennäköisyys ($P(E = 1)$) riippuu ainoastaan siitä, onko henkilöllä aivokasvain vain ei (C). Toisaalta jos tiedetään henkilön veren kalsiumpitoisuus (B) ja se, onko henkilöllä kasvain vai ei (C), voidaan määrätä tajuttomuuskohtauksien todennäköisyys ($P(D = 1)$), joka ei tällöin riipu

[1] Esimerkki on teoksesta [105, sivu 196].

Merkintä	*Tulkinta*
$A = 1$:	Tutkittavalla henkilöllä on aivosyöpä.
$B = 1$:	Tutkittavan henkilön veren kalsiumpitoisuus on noussut epänormaalille tasolle.
$C = 1$:	Tutkittavalla henkilöllä on aivokasvain.
$D = 1$:	Tutkittava henkilö saa tajuttomuuskohtauksia.
$E = 1$:	Tutkittava henkilö kärsii vakavista pääkivuista.

Taulukko 3.1: Lääketieteellisessä esimerkissä käytetyt merkinnät.



Kuva 3.2: Ehdollisen riippumattomuuden vaikutus verkkoesitykseen.

päänsäryn esiintymisestä (E), tai (suoraan) siitä, onko henkilöllä aivosyöpä (A) vai ei. Aivokasvaimen (C) esiintymisen todennäköisyys riippuu vastaavasti suoraan ainoastaan siitä, onko henkilöllä aivosyöpä (A), ei muista tekijöistä.

Yllä mainitut tiedot voidaan toisaalta myös päätellä tilastollisesta aineistosta ilman asiantuntijan apua, mikäli saatavilla on riittävä määrä esimerkkitapauksia. Ehdollisen riippumattomuuden käsitettä soveltaen saatu tieto voidaan nyt koodata seuraavasti:

$$
\begin{aligned}
P(E \mid A, B, C, D) &= P(E \mid C) \\
P(D \mid A, B, C) &= P(D \mid B, C) \\
P(C \mid A, B) &= P(C \mid A)
\end{aligned}
\tag{3.2}
$$

Kun otamme nämä ehdolliset riippumattomuudet huomioon, kuvassa 3.1 esitetty verkko muuttuu kuvassa 3.2 esitettyyn muotoon. Kun edelleen piirrämme verkon hieman eri tavalla, ja merkitsemme kuvaan selvyyden vuoksi kutakin solmua vastaavan muuttujan semanttisen tulkinnan, saamme kuvassa 3.3 esitetyn Bayes-verkkostruktuurin.

Bayes-verkot määritellään formaalisti seuraavasti:

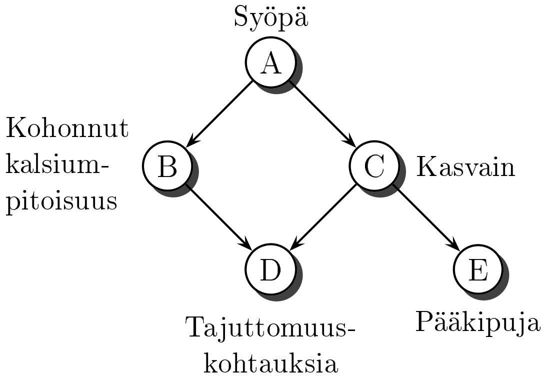
Määritelmä 2 (Bayes-verkkomalli)
Bayes-verkkomalli on pari (M, θ), missä M on syklitön suunnattu verkko, jonka solmut vastaavat ongelmakentän muuttujia, ja jonka topologia täyttää

seuraavan ehdon: kukin muuttuja on ehdollisesti riippumaton kaikista muuttujista, jotka eivät ole muuttujan jälkeläisiä verkossa, annettuna muuttujan edeltäjät, ja muuttujan edeltäjien muodostama joukko on suppein muuttujajoukko joka täyttää tämän ehdon. Parametrijoukon θ muodostavat ehdolliset todennäköisyydet muotoa $P(X \mid F(X))$, missä $F(X)$ on muuttujan X edeltäjien joukko verkossa M.

Tässä yhteydessä on korostettava, että vaikka sekä neuroverkko että Bayes-verkko ovat molemmat verkkoesityksiä, esitysten abstraktiotasossa on huomattava ero. Neuroverkothan ovat lähinnä *funktionaalisen* tason kuvauksia siinä mielessä että neuroverkkoalgoritmien voidaan ajatella toimivan neuroverkkostruktuurin kaltaisessa arkkitehtuurissa, jossa verkon solmut ovat yksinkertaisia laskentayksiköitä jotka lähettävät toisilleen yksittäisistä reaaliluvuista muodostuvia sanomia. Bayes-verkko puolestaan on *käsitteellisen (symbolisen)* tason esitys: Bayes-verkon solmut vastaavat ongelmakentän kuvaamiseen käytettyjä attribuutteja, ja kaaret kuvaavat attribuuttien välisiä riippuvuuksia. Bayes-verkon rakenne ei siis välttämättä suoraan kuvaa Bayes-verkkoalgoritmien toimintaa samalla tavoin kuin neuroverkon rakenne kuvaa neuroverkkoalgoritmien toimintaa (tähän kysymykseen palaamme luvussa 3.2). Erityisesti on huomattava että toisin kuin neuroverkoissa, joissa kuhunkin kaareen liittyy yksi parametri (kaaren paino), Bayes-verkoissa parametrit eivät liity suoranaisesti verkon kaariin, vaan kutakin verkon *solmua* kohden tarvitaan joukko parametreja, joiden lukumäärä on eksponentiaalinen solmun edeltäjien lukumäärän suhteen. Tarvittavien parametrien lukumäärän pienentämiseksi voidaan tätä parametrijoukkoa yrittää approksimoida lineaarisella määrällä parittaisia, muotoa $P(X = x \mid Y = y)$ parametreja, joiden voidaan ajatella liittyvän verkon kaariin: yhtä tällaista approksimaatioita (ns. Noisy-Or-mallia) käsitellään tarkemmin luvussa 3.4.

Kuvassa 3.3 on annettu esimerkki Bayes-verkkomallista kiinnittämällä verkon parametrit lähteessä [105, sivu 197] annettuihin arvoihin (on muistettava että kyseessä on yksinkertaistettu, rajoitettu esimerkki, joten annettuihin parametriarvoihin ei pidä suhtautua lääketieteelliseltä kannalta liian vakavasti). Kaikkien mahdollisten Bayes-verkkomallien (kaikki mahdolliset Bayes-verkkostruktuurit + kaikki mahdolliset struktuuria vastaavien parametrien asetukset) muodostamaa joukkoa kutsutaan seuraavassa *Bayes-verkkomalliperheeksi*. Bayes-verkkojen oppimisella tarkoitetaan yhden Bayes-verkkomallin etsimistä tästä joukosta. Luvussa 3.3 paneudumme tähän kysymykseen tarkemmin.

Bayes-verkkojen määritelmästä ja ehdollisen riippumattomuuden käsitteestä seuraa tärkeä Bayes-verkkomallien avulla koodattujen todennäköisyysjakaumien ominaisuus: jos ongelmakentän kuvauksessa käytetyt muuttujat



$P(A = 1)$	=	0.20
$P(B = 1 \mid A = 1)$	=	0.80
$P(B = 1 \mid A = 0)$	=	0.20
$P(C = 1 \mid A = 1)$	=	0.20
$P(C = 1 \mid A = 0)$	=	0.05
$P(D = 1 \mid B = 1, C = 1)$	=	0.80
$P(D = 1 \mid B = 0, C = 1)$	=	0.80
$P(D = 1 \mid B = 1, C = 0)$	=	0.80
$P(D = 1 \mid B = 0, C = 0)$	=	0.05
$P(E = 1 \mid C = 1)$	=	0.80
$P(E = 1 \mid C = 0)$	=	0.60

Kuva 3.3: Esimerkki Bayes-verkkomallista.

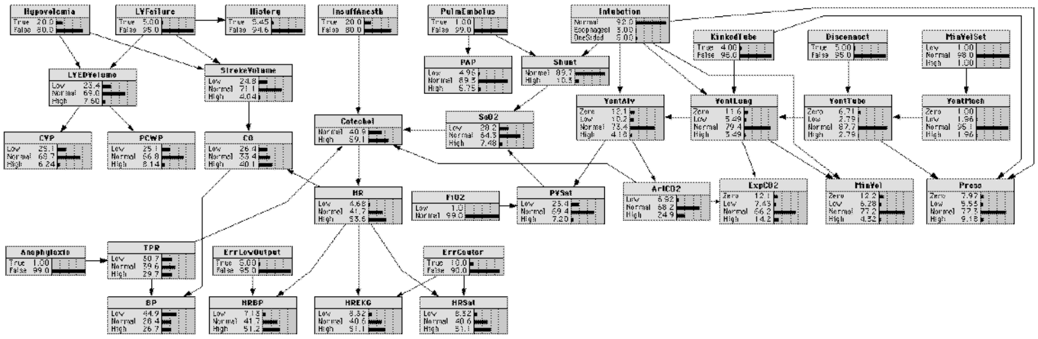
ovat $X_1, \ldots, X_n$, niin minkä tahansa muuttujien arvokombinaation yhteistodennäköisyys saadaan laskettua kaavalla

$$P(X_1 = x_1, \ldots, X_n = x_n) = \prod_{i=1}^{n} P(X_i = x_i \mid F(X_i) = f(X_i)), \qquad (3.3)$$

missä kuten edellä, $F(X_i)$ on muuttujan X_i edeltäjät käytetyssä Bayes-verkossa, ja $f(X_i)$ joukkoa $F(X_i)$ vastaava arvovektori. Näemme, että Bayes-verkkomalli (M, θ) on probabilistinen sillä annetun verkkostruktuurin parametrit kiinnittävät yhteistodennäköisyysjakauman muuttujille $X_1, \ldots, X_n$. Luvussa 3.2 esitamme kuinka hajotelmaa (3.3) voidaan käyttää hyväksi toteutettaessa luvussa 1.3 esitettyjä probabilistisen päättelyn muotoja.

Kuvassa 3.3 esitetyssä Bayes-verkkomallissa on tarvittavien parametrien lukumäärä 11, mikä on edelleen yli 30% koko avaruuden koosta. On kuitenkin huomattava, että tarvittavat parametrit, kuten esimerkiksi $P(B|A)$, ovat muodoltaan sangen yksinkertaisia, ja siten myös helposti asiantuntijoiden määrättävissä, toisin kuin käsitteellisesti hankalat yhteistodennäköisyydet $P(A, B, C, D, E)$. Lisäksi on korostettava että tarvittavien parametrien suhde avaruuden kokoon pienenee dramaattisesti muuttujien lukumäärän kasvaessa: kuvassa 3.4 on sairaalapotilaiden monitoroinnissa käytetty 37-solmuinen ALARM-verkkona tunnettu Bayes-verkkostruktuuri (katso [8]), jossa tarvittavien parametrien (ei näytetty kuvassa) lukumäärä on 509, ainoastaan $2,94 \times 10^{-12}$% koko avaruuden koosta ($2^{13} \times 3^{17} \times 4^7 = 1,73 \times 10^{16}$). Kuvassa esitetyt todennäköisyydet ovat eri muuttujien arvojen prioritodennäköisyyksiä, jotka saadaan laskettua ALARM-verkon parametreista. Kuva on tuotettu Norsys-yhtiön Netica-ohjelmistolla, joka on Bayes-verkkojen konstruoimiseen ja niiden käyttämiseen tarkoitettu sovelluskehitin (katso luku 4.2).

Käytännön sovellusten kannalta on kuvassa 3.4 esitetty Bayes-verkkokin



Kuva 3.4: Sairaalapotilaiden valvonnassa käytettävä ALARM-verkko.

edelleen melko pieni — Bayes-verkko-ohjelmistoja valmistava HUGIN-yhtiö ilmoittaa teollisessa käytössä olevan yli tuhat solmua sisältäviä Bayes-verkkoja. Tässä yhteydessä on muistettava, että toisin kuin neuroverkoissa, ei Bayes-verkoissa solmujen lukumäärä ole vapaa parametri, vaan se määräytyy ongelmakentän kuvaamisessa käytettävien attribuuttien lukumäärästä: jo sadallakin solmulla (attribuutilla) voidaan kuvata sangen monimutkaisia ongelmakenttiä. Toisaalta on korostettava, että vaikka solmujen lukumäärä on kiinnitetty, ei Bayes-verkkojen konstruoiminen ole kuitenkaan välttämättä yksinkertaista, koska mahdollisten verkkotopologioiden lukumäärä kasvaa erittäin nopeasti solmujen lukumäärän suhteen (katso luku 3.3).

3.1.1 Bayes-verkot ja kausaalisuus

Kuvassa 3.3 esitetyssä esimerkissä voidaan Bayes-verkkostruktuurin kaarille antaa selkeä kausaalinen tulkinta: esimerkiksi muuttujan A, joka on muuttujien B ja C edeltäjä verkossa, voidaan ajatella olevan muuttujien B ja C kausaalinen *syy*. Mielenkiintoinen kysymys onkin kuinka Bayes-verkkostruktuurin kaaret suhtautuvat kausaalisen syy-seuraussuhteen käsitteeseen — onko kaikkien Bayes-verkkojen kaarilla sama kausaalinen tulkinta?

Yleisesti ottaen vastaus tähän kysymykseen on kielteinen: Bayes-verkon kaarien ei välttämättä tarvitse kuvastaa kausaalisia syy-seuraussuhteita. Tätä seikkaa havainnollistaa seuraava yksinkertainen esimerkki: olkoon ongelmakentän kuvauksessa käytettävät muuttujat A ja B, joilla on seuraavat semanttiset tulkinnat:

A: Tutkittavalla henkilöllä on flunssa (A=1) tai ei (A=0).

B: Tutkittavan henkilön nenä vuotaa (B=1) tai ei (B=0).

Vaihtoehtoisia Bayes-verkkostruktuureja on kaksi: $A \rightarrow B$ ja $A \leftarrow B$. Tarkastellaan ensiksi ensimmäistä vaihtoehtoa, ja oletetaan että verkkostruktuuria vastaavat parametrit ovat

$$\begin{aligned} P(B=1 \mid A=1) &= 0.8, \\ P(B=1 \mid A=0) &= 0.01, \text{ ja} \\ P(A=1) &= 0.001. \end{aligned}$$

Nämä määrittelevät seuraavan yhteisjakauman:

$$\begin{aligned} &P(A=0,B=0) = P(A=0)P(B=0|A=0) = 999/1000 * 99/100 = 0.98901, \\ &P(A=0,B=1) = P(A=0)P(B=1|A=0) = 999/1000 * 1/100 = 0.00999, \\ &P(A=1,B=0) = P(A=1)P(B=0|A=1) = 1/1000 * 2/10 = 0.0002, \text{ja} \\ &P(A=1,B=1) = P(A=1)P(B=1|A=1) = 1/1000 * 8/10 = 0.0008. \end{aligned}$$

Toisaalta, jos otamme verkkostruktuurivaihtoehdon $A \leftarrow B$, ja asetamme struktuuria vastaavat parametrit seuraavasti:

$$\begin{aligned} P(B=1) &= 0.98921, \\ P(A=1 \mid B=1) &= 0.074143, \\ P(A=1 \mid B=0) &= 0.000202, \end{aligned}$$

voidaan yksinkertaisella laskutoimituksella osoittaa että syntyvä yhteistodennäköisyysjakauma on täsmälleen sama kuin ensimmäisessä tapauksessa. Esitettyjen verkkostruktuurien sanotaan olevan *ekvivalentteja*, mikä tarkoittaa sitä että niillä voidaan esittää täsmälleen sama joukko todennäköisyysjakaumia.

Kumpi kahdesta ekvivalentista verkkostruktuurista on sitten "oikea", kumpaa tulisi käyttää käytännön sovelluksissa? Puhtaasti pragmaattiselta kannalta katsottuna tällä kysymyksella ei ole merkitystä: koska syntyvät mallit esittävät täsmälleen samoja todennäköisyysjakaumia, ne käyttäytyvät täsmälleen samalla tavalla. Bayes-verkkojen soveltamisen kannalta on siis tässä mielessä yhdentekevää, käytetäänkö esimerkissämme verkkostruktuuria $A \rightarrow B$, joka vastannee useimpien käsitystä muuttujien kausaalisesta syy-seuraus-riippuvuussuhteesta, vai verkkostruktuuria $A \leftarrow B$. Bayes-verkkojen kaaret eivät siis välttämättä vastaa kausaalisia syy-seuraussuhteita.

Toisaalta on kuitenkin todettava, että vaikka kaarien kausaalisella tulkinnalla ei ole merkitystä Bayes-verkkojen soveltamisen kannalta, saattaa kausaalisuuden käsitteellä olla suuri käytännön merkitys Bayes-verkkojen rakentamisprosessissa: on helppo nähdä että muotoa $P(\text{seuraus} \mid \text{syy})$ olevien todennäköisyyksien arviointi on huomattavasti helpompaa kuin muotoa $P(\text{syy} \mid \text{seuraus})$ olevien todennäköisyyksien. Esimerkkitapauksessamme on varmaankin huomattavasti helpompi arvioida mikä on nenän valumisen todennäköisyys, annettuna että henkilöllä on flunssa, kuin arvioida todennäköisyyttä sille että henkilöllä on flunssa, annettuna että hänen nenänsä valuu. Samoin flunssan esiintymisen prioritodennäköisyyden arvioiminen on helpompaa kuin nenän valumisen prioritodennäköisyyden valuminen, koska jälkimmäisessä tapauksessa on arvioitava kaikki mahdolliset tilanteet joissa nenän valumista voi esiintyä. Jos malli siis konstruoidaan asiantuntijatietämystä käyttäen, on käytännössä järkevää pyrkiä rakentamaan verkko siten että solmujen väliset kaaret vastaavat muuttujien kausaalisia riippuvuussuhteita.

Vaikka kysymyksellä Bayes-verkkojen kaarien kausaalisesta tulkinnasta ei ole käytännön merkitystä jos malleja sovelletaan probabilistiseen päättelyyn, on kysymys sitäkin tärkeämpi ongelmakentän ominaisuuksien tilastollisen analyysin (data mining) kannalta. Bayes-verkkoja, joissa kaikki verkon kaaret kuvaavat kausaalisia riippuvuussuhteita, kutsutaan *kausaaliverkoiksi (causal networks)*. Bayes-verkkotutkimus onkin avannut täysin uusia lähtökohtia kausaalisuuden tutkimukselle sinänsä (katso esim. [106, 130, 47, 52]).

3.2 Päättely Bayes-verkoissa

Kuten luvussa 1.3 näimme, on probabilistisessa päättelyssä tavoitteena estimoida muotoa $P(\mathbf{S}_1 \mid \mathbf{S}_2 = \mathbf{s}_2, M, \theta)$ olevia todennäköisyyksiä joko niin, että tuloksena on yhteisjakauma joukon $\mathbf{S}_1$ muodostaville muuttujille, tai niin että tuloksena saadaan n todennäköisintä arvokombinaatiota $\mathbf{s}_1$. Esittelemme seuraavassa kuinka tämä toteutetaan Bayes-verkkojen tapauksessa. Yksinkertaisuuden vuoksi rajoitumme aluksi esimerkissä 9 kuvattuun tapaukseen, jossa joukon $\mathbf{S}_1$ koko on yksi, eli tarkastelemme muotoa $P(X = x \mid \mathbf{S}_2 = \mathbf{s}_2, M, \theta)$ olevia todennäköisyyksiä. Yleistys monimutkaisempiin tapauksiin on suoraviivainen.

Esimerkissä 9 kuvattua ongelmakenttää voidaan mallintaa kuvassa 3.5 esitetyllä Bayes-verkolla, kun formalisoimme esimerkin kuvauksessa annetut oletukset seuraavasti: nenän vuotamiseen (muuttuja C) vaikuttavat muuttujat ovat A (onko flunssa vai ei) ja B (onko talossa kissa vai ei). Huonekaluista löytyvien raapimisjälkiin (muuttuja D) vaikuttaa se, onko talossa kissa, eivät muut mallissa käytetyt attribuutit.

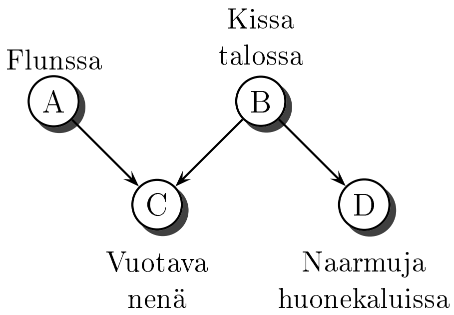
Mallimaailmassa on kaikkiaan 16 mahdollista tilaa, joita vastaavat yhteistodennäköisyydet saadaan laskettua esitetyn Bayes-verkkomallin avulla soveltamalla kaavaa (3.3), josta näemme että

$$P(A, B, C, D) = P(A)P(B)P(C \mid A, B)P(D \mid B).$$

Esimerkki 9 Kuvassa 3.5 esitettyyn Bayes-verkkoon liittyvän ongelmakentän kuvaus.

Olet matkalle vierailulle kun muistat äkkiä että et muistanut ottaa nenäliinoja mukaan vaikka kaupungissa on paha influenssaepidemia (arvioit tartunnan todennäköisyyden suuruudeksi 0.05). Olet myös allerginen kissoille, etkä muista varmasti onko isäntäperheellä kissa vai ei. Arvioit todennäköisyydeksi sille että talossa on kissa 0.05. Olet huolestunut koska sekä flunssa että kissa-allergia saa nenäsi yleensä vuotamaan, jolloin nenäliinoille olisi ollut tarvetta. Kokemuksesta tiedät että jos sinulla on joko flunssa, tai olet kosketuksissa kissojen kanssa, nenäsi alkaa vuotaa todennäköisyydellä 0.95, ja arvelet että jos molemmat pitäisivät yhtäaikaa paikkansa, nenän vuotamisen todennäköisyys nousisi arvoon 0.99. Toisaalta, ilman edellä mainittuja syitä on äärimmäisen epätodennäköistä (todennäköisyysarviosi on 0.001) että nenäsi vuotaisi.

Päätät perille tultuasi ensitöikseksi tarkkailla näkyykö huonekaluissa kissan tekemiä naarmuja, joita kokemuksesi syntyy todennäköisyydellä 0.99, mikäli talossa asuu kissa. On tietysti myös mahdollista että huonekaluista löytyy kissan tekemiä naarmuja, vaikka talossa ei kissaa olisikaan (huonekaluthan voivat olla vaikkapa ostettu käytettyinä), mutta arvioit tämän mahdollisuuden todennäköisyydeksi ainoastaan 0.1.



$P(A = 1) = 0.05$
$P(B = 1) = 0.05$
$P(C = 1 \mid A = 0, B = 0) = 0.001$
$P(C = 1 \mid A = 1, B = 0) = 0.95$
$P(C = 1 \mid A = 0, B = 1) = 0.95$
$P(C = 1 \mid A = 1, B = 1) = 0.99$
$P(D = 1 \mid B = 1) = 0.99$
$P(D = 1 \mid B = 0) = 0.1$

Kuva 3.5: Esimerkin 9 Bayes-verkkokuvaus.

Vastaavat 16 yhteistodennäköisyyttä on esitetty taulukossa 3.2.

Taulukossa 3.2 esitettyjä yhteistodennäköisyyksiä hyväksikäyttäen voidaan laskea (priori)todennäköisyys sille että nenä alkaa vuotaa esimerkissä 9

A	B	C	D	P(A)	P(B)	P(C|A,B)	P(D|B)	P(A,B,C,D)
1	1	1	1	0.05	0.05	0.99	0.99	2.450E-03
1	1	1	0	0.05	0.05	0.99	0.01	2.475E-05
1	1	0	1	0.05	0.05	0.01	0.99	2.475E-05
1	1	0	0	0.05	0.05	0.01	0.01	2.500E-07
1	0	1	1	0.05	0.95	0.95	0.1	4.513E-03
1	0	1	0	0.05	0.95	0.95	0.9	4.061E-02
1	0	0	1	0.05	0.95	0.05	0.1	2.375E-04
1	0	0	0	0.05	0.95	0.05	0.9	2.138E-03
0	1	1	1	0.95	0.05	0.95	0.99	4.467E-02
0	1	1	0	0.95	0.05	0.95	0.01	4.513E-04
0	1	0	1	0.95	0.05	0.05	0.99	2.351E-03
0	1	0	0	0.95	0.05	0.05	0.01	2.375E-05
0	0	1	1	0.95	0.95	0.001	0.1	9.025E-05
0	0	1	0	0.95	0.95	0.001	0.9	8.123E-04
0	0	0	1	0.95	0.95	0.999	0.1	9.016E-02
0	0	0	0	0.95	0.95	0.999	0.9	8.114E-01
								1.000E+00

Taulukko 3.2: Kuvassa 3.5 esitetyn Bayes-verkon määrittelemä yhteistodennäköisyysjakauma.

esitetyssä mallissa:

$$\begin{aligned}
P(C=1) &= \sum_{a\in\{0,1\}} \sum_{b\in\{0,1\}} \sum_{d\in\{0,1\}} P(C=1, A=a, B=b, D=d) \\
&= 2.450\text{E-}03 + 2.475\text{E-}05 + 4.513\text{E-}03 + 4.061\text{E-}02 \\
&\quad +4.467\text{E-}02 + 4.513\text{E-}04 + 9.025\text{E-}05 + 8.123\text{E-}04 = 9.36\%.
\end{aligned}$$

Vastaavasti saamme naarmujen esiintymisen (priori)todennäköisyyden $P(D = 1)$ arvoksi 14.45%. Flunssan (priori)todennäköisyys $P(A = 1)$, samoin kuin kissan esiintymisen (priori)todennäköisyys $P(B = 1)$ on 5%, mitkä nähdään suoraan käytetystä Bayes-verkkomallista.

Sana "priori" viittaa edellä todennäköisyyteen, joka saadaan Bayes-verkkomallista joko suoraan tai yksinkertaisen laskutoimituksen kautta, ilman että tehdyt havainnot vaikuttavat tilanteeseen. Probabilististen mallien käyttökelpoisuus perustuu suoraviivaiseen kalkyyliin, jonka avulla eri satunnaismuuttujien todennäköisyydet muuttuvat dynaamisesti sitä mukaa kun saamme ongelmakentästä uusia havaintoja. Kun havaitset esimerkiksi että nenäsi alkaa vuotaa ($C = 1$), voit *päivittää* uskomuksesi siihen että sinulla on flunssa laskemalla flunssan ehdollisen todennäköisyyden *annettuna tehdyt havainnot*:

$$\begin{aligned}P(A=1 \mid C=1) &= \frac{P(A=1, C=1)}{P(C=1)} \\ &= \frac{\sum_{b\in\{0,1\}}\sum_{d\in\{0,1\}} P(A=1, B=b, C=1, D=d)}{P(C=1)} \\ &= \frac{2.450\text{E-}03 + 2.475\text{E-}05 + 4.513\text{E-}03 + 4.061\text{E-}02}{9.36\text{E-}02} \\ &= 50.84\%.\end{aligned}$$

Flunssan todennäköisyys siis yli kymmenkertaistui tehdyn havainnon seurauksena. Vastaavasti saamme todennäköisyydeksi sille että talossa on kissa, annettuna se havainto että nenäsi vuotaa,

$$P(B=1 \mid C=1) = 50.84\%,$$

ja todennäköisyydeksi sille että huonekaluissa on naarmuja

$$P(D=1 \mid C=1) = 55.25\%.$$

Edellä näimme että annettuna havainto, “nenä vuotaa”, saamme sekä flunssan että kissan esiintymisen todennäköisyydeksi 0.5084 ja naarmujen esiintymisen todennäköisyydeksi 0.5525. Jos kysymme nyt mikä on todennäköisin mallimaailman tila, annettuna havainto $C = 1$, intuitiivisesti houkuttelevalta vaihtoehdolta tuntuisi asettaa kukin binäärimuuttuja itsenäisesti siihen arvoon jonka todennäköisyys on suurempi, jolloin vastaus olisi

$$P(A=1, B=1, C=1, D=1) = 2.450\text{E-}03.$$

Taulukosta 3.2 on kuitenkin helppo nähdä että muuttujien konfiguraatioavaruudesta löytyy kolme seikan $C = 1$ kanssa konsistenttia tilaa, joiden todennäköisyys on suurempi:

$$\begin{aligned}P(A=0, B=1, C=1, D=1) &= 4.467\text{E-}02, \\ P(A=1, B=0, C=1, D=0) &= 4.061\text{E-}02, \text{ ja} \\ P(A=1, B=0, C=1, D=1) &= 4.513\text{E-}03.\end{aligned}$$

Kokonaistodennäköisuuden maksimoivan tilan etsiminen on siis aivan eri ongelma kuin kunkin yksittäisen satunnaismuuttujan ehdollisen todennäköisyysjakauman laskeminen.

Kuten edellä esitetystä esimerkistä näimme, mikä tahansa Bayes-verkkoesitystä vastaavan yhteistodennäköisyysjakauman arvo saadaan laskettua

Selitysten poissulkeminen (explaining away)
Annettuna nenän vuotamisesta tehty havainto saamme sekä flunssan että kissan esiintymisen todennäköisyydeksi 50.84%, siis vain hieman yli 0.5. Ensi näkemältä voi tuntua siltä, että tehty havainto ei ole merkittävästi lisännyt tietoamme ongelmakentästä, mutta täytyy muistaa että sekä flunssan että kissan esiintymisen prioritodennäköisyys oli mallissamme hyvin pieni, vain 5%, joten näiden muuttujien todennäköisyys on yli kymmenkertaistunut. Toisaalta näemme että tehty havainto ei tässä vaiheessa auta meitä päättämään, kumpi seikka on nenän valumisen syy, flunssa vai kissa-allergia. Jos havaitsemme edelleen että talon huonekaluissa on kissan tekemiä naarmuja, satunnaismuuttujien posterioritodennäköisyydet päivittyvät jälleen:

$$
\begin{aligned}
P(B = 1 \mid C = 1, D = 1) &\approx 91\% \\
P(A = 1 \mid C = 1, D = 1) &\approx 13\%
\end{aligned}
$$

Tässä uudessa tilanteessa voimme olettaa että nenän vuotamisen aiheuttaja on melko varmasti kissa-allergia. Tämä esimerkki havainnollistaa todennäköisyysmalleissa esiintyvää havaintojen selityksiin liittyvää *"poisselittämis" (explaining away)*-ilmiötä: yhden vaihtoehtoisen selityksen todennäköisyyden kasvaminen vähentää automaattisesti vaihtoehtoisten selitysten uskottavuutta, vaikka selitykset eivät sinänsä olisikaan poissulkevia. Tämän intuitiivisesti luontevalta tuntuvan periaatteen on havaittu esiintyvän myös inhimillisessä päättelyssä [55, 104, 87].

kertomalla asiaankuuluvat Bayes-verkon parametrit keskenään. Näin ollen probabilistisen päättelyn suorittamiseksi ei tarvitse tallettaa taulukon 3.2 kaltaista todennäköisyystaulukkoa, vaan riittää tallettaa Bayes-verkon parametrit, joita on yleensä pieni määrä koko malliavaruuden kokoon verrattuna, kuten edellisessä luvussa näimme. Toisalta probabilistisessa päättelyssä käytettävien ehdollisten todennäköisyyksien laskeminen edellyttää yllä esitetyssä muodossaan monia tekijöitä sisältävien marginalisointisummien laskemista, mikä saattaa suurten Bayes-verkkojen tapauksessa olla käytännössä mahdotonta. Bayes-verkkojen määritelmään liittyvät ehdolliset riippumattomuusoletukset tarjoavat kuitenkin ratkaisun myös tähän ongelmaan, kuten seuraava esimerkki osoittaa.

Olettakaamme että tehtävänä on laskea kuvassa 3.5 esitettyyn Bayes-verkkomalliin liittyen ehdollinen todennäköisyys $P(B = 1 \mid C = 1, D = 1)$.

Todennäköisyyslaskennan perusaksioomia käyttäen näemme helposti että

$$P(B=1 \mid C=1, D=1) = \frac{P(d^+ \mid b^+, c^+)P(c^+ \mid b^+)P(b^+)}{P(c^+, d^+)},$$

missä arvoasetuksesta $X=1$ on käytetty lyhennettä x^+ (asetuksesta $X=0$ käytetään jatkossa lyhennettä x^-). Bayes-verkkoihin liittyvän ehdollisen riippumattomuuden määritelmän perustella muuttuja D on ehdollisesti riippumaton muuttujasta C, annettuna muuttujan B arvo, joten saamme

$$P(b^+ \mid c^+, d^+) = \frac{P(d^+ \mid b^+)P(c^+ \mid b^+)P(b^+)}{P(c^+, d^+)}.$$

Toisaalta, koska

$$P(b^+ \mid c^+, d^+) = \frac{P(b^+ \mid c^+, d^+)}{P(b^+ \mid c^+, d^+) + P(b^- \mid c^+, d^+)},$$

saamme että

$$\begin{aligned} P(b^+ \mid c^+, d^+) &= \frac{\frac{P(d^+|b^+)P(c^+|b^+)P(b^+)}{P(c^+,d^+)}}{\frac{P(d^+|b^+)P(c^+|b^+)P(b^+)}{P(c^+,d^+)} + \frac{P(d^+|b^-)P(c^+|b^-)P(b^-)}{P(c^+,d^+)}} \\ &= \frac{P(d^+ \mid b^+)P(c^+ \mid b^+)P(b^+)}{P(d^+ \mid b^+)P(c^+ \mid b^+)P(b^+) + P(d^+ \mid b^-)P(c^+ \mid b^-)P(b^-)}. \end{aligned}$$

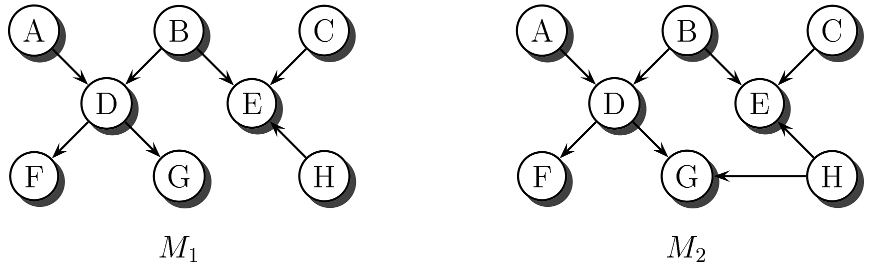
Näemme että koska nimittäjissä esiintyvä termi $P(c^+, d^+)$ ei riipu muuttujan B arvosta (eli on B:n ehdollisen todennäköisyysjakauman suhteen vakio), sen arvoa ei tarvitse laskea, vaan se kumoutuu pois! Haluamamme ehdollisen todennäköisyyden saamiseksi riittää siis laskea kaksi tuloa,

$$\begin{aligned} &P(d^+ \mid b^+)P(c^+ \mid b^+)P(b^+) \text{ ja} \\ &P(d^+ \mid b^-)P(c^+ \mid b^-)P(b^-), \end{aligned}$$

jotka molemmat sisältävät vain Bayes-verkkomalliin talletettuja parametreja.

Bayes-verkoissa tapahtuvan probabilistisen päättelyn havainnollistamiseksi ajatellaan kuvan 3.5 Bayes-verkkoa tietokoneohjelma-arkkitehtuurina, jossa kukin muuttuja X vastaa moduulia, jonka tehtävänä on laskea ehdollinen todennäköisyysjakauma $P(X \mid \mathbf{S}_2 = \mathbf{s}_2)$, ja oletetaan että Bayes-verkon parametrit on talletettu siten, että kukin moduuli X näkee muotoa $P(X \mid F(X))$ olevat parametrit. Kuten edellä näimme, yllä esitetyssä tapauksessa moduuli B tarvitsee seuraavat luvut voidakseen määrätä ehdollisen todennäköisyysjakauman $P(b^+ \mid c^+, d^+)$:

$$P(d^+ \mid b^+), P(c^+ \mid b^+), P(b^+), P(d^+ \mid b^-), P(c^+ \mid b^-), \text{ ja} P(b^-).$$



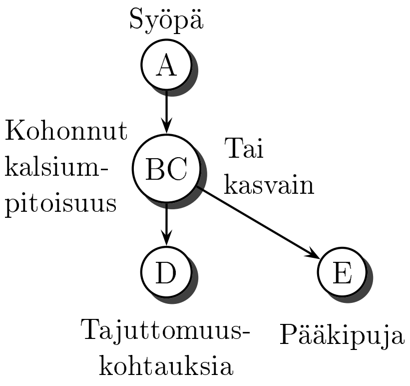
Kuva 3.6: Yksipolkuinen (M_1) ja monipolkuinen (M_2) Bayes-verkkostruktuuri.

Luvut $P(b^+)$ ja $P(b^-)$ on talletettu paikallisesti moduuliin B Bayes-verkkomallin parametreina. Koska $P(c^+ \mid b^+) = P(c^+ \mid a^-, b^+) + P(c^+ \mid a^+, b^+)$, moduli C voi laskea tämän tarvittavan luvun, samoin kuin luvun $P(c^+ \mid b^-)$, nopeasti ja tehokkaasti käyttäen vain paikallisesti talletettua informaatiota (moduuliin C talletettuja Bayes-verkon parametreja). Moduuli D voi vastaavasti laskea arvot $P(d^+ \mid b^+)$ ja $P(d^+ \mid b^-)$. Laskeakseen todennäköisyyden $P(b^+ \mid c^+, d^+)$ modulin B riittää siis lähettää tarvittavien lukujen laskupyyntö moduleille C ja D, ja kertoa saadut vastaukset edellä esitetyllä tavalla halutun ehdollisen todennäköisyysjakauman tuottamiseksi.

Edellä kuvattua viestinvälitysarkkitehtuuriin perustuva menetelmää ehdollisten todennäköisyyksien laskemiseksi voidaan soveltaa kaikissa *yksipolkuisissa (singly connected)* Bayes-verkoissa: yksipolkuisissa verkoissa ei minkään kahden solmun välillä ole kuin yksi suuntaamaton polku, missä suuntaamaton polku tarkoittaa verkon kaaria seuraavaa reittiä, joka voi kulkea suunnattuja kaaria kumpaan suuntaan tahansa. Kuvassa 3.6 yksipolkuinen Bayes-verkkostruktuuri M_1 muutetaan *monipolkuiseksi (multi-connected)* yhdistämällä solmut H ja G, jolloin syntyy kaksi (suuntaamatonta) polkua esimerkiksi solmujen B ja G välille.

Yksipolkuisissa Bayes-verkoissa voidaan ehdollisten todennäköisyysjakaumien laskeminen suorittaa erittäin tehokkaasti, lineaarisessa ajassa Bayes-verkon parametrien lukumäärään nähden. Algoritmin yksityiskohtainen kuvaus löytyy esimerkiksi alan perusoppikirjoista [105, 97]. Maksimitodennäköisyyskonfiguraation etsiminen voidaan niinikään suorittaa samankaltaisella algoritmilla.

Edellä kuvatunlaiseen viestinvälitystekniikkaan perustuvat tehokkaat probabilistiset päättelyalgoritmit perustuvat siihen tosiseikkaan, että yksipolkuisissa verkoissa kukin solmu osittaa verkon kahteen erilliseen osaan, jotka voidaan käsitellä erikseen, ja sama pätee rekursiivisesti jokaiselle verkon solmulle. Monipolkuisissa verkoissa menetelmä ei valitetavasti toimi. Itse asiassa voidaan teoreettisesti osoittaa, että muotoa $P(X = 1 \mid \mathbf{S}_2 = \mathbf{s}_2)$ olevien



$P(A=1)$	$=$	0.20
$P(BC=00 \mid A=1)$	$=$	$0.20 \times 0.80 = 0.16$
$P(BC=01 \mid A=1)$	$=$	$0.20 \times 0.20 = 0.04$
$P(BC=10 \mid A=1)$	$=$	$0.80 \times 0.80 = 0.64$
$P(BC=11 \mid A=1)$	$=$	$0.80 \times 0.20 = 0.16$
$P(BC=00 \mid A=0)$	$=$	$0.80 \times 0.95 = 0.76$
$P(BC=01 \mid A=0)$	$=$	$0.80 \times 0.05 = 0.40$
$P(BC=10 \mid A=0)$	$=$	$0.20 \times 0.95 = 0.19$
$P(BC=11 \mid A=0)$	$=$	$0.20 \times 0.05 = 0.01$
$P(D=1 \mid BC=11)$	$=$	0.80
$P(D=1 \mid BC=01)$	$=$	0.80
$P(D=1 \mid BC=10)$	$=$	0.80
$P(D=1 \mid BC=00)$	$=$	0.05
$P(E=1 \mid BC=01)$	$=$	0.80
$P(E=1 \mid BC=11)$	$=$	0.80
$P(E=1 \mid BC=00)$	$=$	0.60
$P(E=1 \mid BC=10)$	$=$	0.60

Kuva 3.7: Kuvan 3.3 Bayes-verkko muutettuna yksipolkuiseksi verkoksi.

ehdollisten todennäköisyyksien laskeminen [27] tai edes niiden approksimointi [29, 115], ja myös maksimitodennäköisyystilojen etsiminen [124], on erittäin vaikea ongelma monipolkuisten verkkojen tapauksessa: jos näille ongelmille olisi aina polynomisessa ajassa toimiva ratkaisumenetelmä, menetelmää voitaisiin käyttää ratkaisemaan kaikki ns. NP-täydelliset päätösongelmat polynomisessa ajassa, minkä ei yleisesti uskota olevan mahdollista. On kuitenkin syytä muistaa näiden teoreettisten tulosten käsittelevän pahimman tapauksen (worst-case) skenaariota: yhdenkin eksponentiaalisen ajan vaativan tapauksen löytäminen riittää, vaikka kaikki muut tapaukset voitaisiin käsitellä polynomisessa ajassa.

Yleisin lähestymistapa probabilistisen päättelyn toteuttamiseksi monipolkuisissa Bayes-verkoissa on yrittää muuttaa monipolkuinen verkko yksipolkuiseksi, jolloin päättely on tehokasta kuten edellä näimme. Yleisimmin käytetty transformointitekniikka on muuttujien *ryhmittely (clustering)*. Tarkastellaan esimerkiksi kuvassa 3.3 esitettyä Bayes-verkkoa. Ryhmittelemällä muuttujat uudelleen siten että ne muodostavat neljä ryhmää $\{A\}$, $\{B, C\}$, $\{D\}$, ja $\{E\}$ näemme, että jos sulautamme muuttujat B ja C uudeksi muuttujaksi BC, syntyvä Bayes-verkko on yksipolkuinen (katso kuva 3.7). Koska muuttujilla B ja C oli kummallakin 2 mahdollista arvoa, on yhdistetyllä muuttujalla BC 4 mahdollista arvoa; merkitään näitä arvoja symboleilla 00,01,10 ja 11. Syntyneeseen uuteen verkkoon liittyvät todennäköisyydet on helppo laskea alkuperäisen verkon parametriarvoista: muotoa $P(D \mid BC)$ olevat todennäköisyydet vastaavat suoraan muotoa $P(D \mid B, C)$ olevia todennäköisyyksiä, ja ehdollisen riippumattomuuden määritelmän mukaan näemme että $P(E \mid BC) = P(E \mid C)$ ja $P(BC \mid A) = P(B \mid A)P(C \mid A)$.

Edellä kuvatussa yksinkertaisessa tapauksessa oli helppo nähdä kuinka annettu monipolkuinen verkko voitiin muuttaa yksipolkuiseksi yhdistämällä kaksi muuttujaa, mutta monimutkaisemmissa verkoissa tehtävä on huomattavasti vaikeampi eivätkä ratkaisut ole aina yhtä yksinkertaisia. Käytännön kannalta tärkeä kysymys on voidaanko tämä transformaatioprosessi automatisoida — onko mahdollista kehittää algoritmi joka tuottaa mistä tahansa annetusta monipolkuisesta Bayes-verkosta yksipolkuisen Bayes-verkon polynomisessa ajassa? Vastaus on hieman yllättäen kyllä: tällaisia algoritmeja on kehitetty itse asiassa lukuisia (katso esim. [105, 97, 60, 18]). Useimmat esitetyistä menetelmistä perustuvat kaksivaiheiseen algoritmiin, jossa ensimmäisessä vaiheessa annettu Bayes-verkko *moralisoidaan*[2] yhdistämällä solmut joilla on yhteinen seuraaja, ja poistamalla kaarien suunnat. Menetelmän toisessa vaiheessa syntynyt suuntaamaton verkko *kolmioidaan (triangulation)* lisäämällä verkkoon suuntaamattomia kaaria kunnes kaikissa verkossa sijaitsevien yli kolmen kaaren pituisissa silmukoissa on joitakin kahta silmukan solmua yhdistävä kaari. Jos kolmioidussa verkossa esiintyvien maksimaalisten *klikkien* (suurimpien sellaisten verkosta löytyvien solmujoukkojen joissa kaikki solmut on kytketty toisiinsa) solmut yhdistetään yhdeksi muuttujaksi, niin syntyvien yhdistettyjen muuttujien avulla on mahdollista luoda alkuperäisen monipolkuisen Bayes-verkon kanssa ekvivalentti yksipolkuinen verkko. Syntyvää yksipolkuista verkkoa kutsutaan *liittymäpuuksi (junction tree)*. Nimitys johtuu siitä, että liittymäpuun solmusta voi olla kaari toiseen solmuun vain silloin jos solmuja vastaavien klikkien leikkaus ei ole tyhjä. Liittymäpuiden konstruoimisessa käytettävä algoritmi on melko monimutkainen, eikä sen esittäminen tämän raportin yhteydessä ole mielekästä. Algoritmin ksityiskohdat selviävät mm. lähteistä [105, 97, 60, 18].

Liittymäpuihin perustuvat menetelmät antavat siis algoritmin probabilistisen päättelyn suorittamiseksi monipolkuisissa Bayes-verkoissa. Koska syntyvä liittymäpuu on puu, siis yksipolkuinen Bayes-verkko, voidaan probabilistista päättelyä suorittaa syntyneessä verkossa lineaarisessa ajassa verkon parametrien lukumäärään nähden. Tässä yhteydessä on kuitenkin huomattava että syntyvän liittymäpuun parametrien lukumäärä ei ole sama kuin alkuperäisen monipolkuisen Bayes-verkon parametrien lukumäärä, vaan se voi olla paljon suurempi, pahimmassa tapauksessa eksponentiaalisen suuri alkuperäiseen lukumäärään verrattuna. Voidaan itse asiassa osoittaa [142] että sellaisen kolmioinnin löytäminen joka tuottaa minimaalisen määrän liittymäpuuparametreja on NP-täydellinen ongelma. Alkuperäisen ongelman vaikeutta ei siis voi paeta tiedon esitystapaa muuttamalla (yhdistämällä muuttujia), mikä on tietenkin muutenkin itsestään selvää. Becker ja Geiger [7] ovat esittä-

[2]Moralisointi-termi (moralization) johtuu siitä että prosessissa yhdistetään solmut (vanhemmat) joilla on yhteinen seuraaja (lapsi).

neet "hajota-ja-hallitse"-periaatteeseen perustuvan algoritmin, jonka tuottama kolmiointi on aina vähintään tietyn vakioetäisyyden päässä optimaalisesta. Kjærulff [66] on tutkinut stokastisten optimointmenetelmien soveltamista tämän ongelman ratkaisemisessa.

Käytännön sovelluksista saadut kokemukset osoittavat että liittymäpuihin perustuvat algoritmit toimivat yleensä varsin hyvin — useimmat luvussa 4.2 kuvatuista ohjelmistoista perustuvat tähän lähestymistapaan. Pearlin *ehdollistamismenetelmä (conditioning)* [104] on läheistä sukua solmujen yhdistämiseen perustuville tekniikoille: ehdollistamisessa monipolkuista verkkoa ei kuitenkaan eksplisiittisesti muuteta yksipolkuiseksi verkoksi, vaan ongelmia aiheuttavat polut poistetaan implisiittisesti kiinnittämällä (ehdollistamalla) polulla sijaitsevia muuttujia johonkin arvoon. Probabilistisen päättelyn toteuttamiseksi voidaan myös luopua laskennan deterministisyydestä, ja käyttää stokastisia päättelyalgoritmeja. Teoksessa [93] kuvataan kuinka kuinka tämän ongelman ratkaisemisessa voidaan käyttää Boltzmannin koneena tunnettua stokastista neuroverkkoarkkitehtuuria. Muita vaihtoehtoisia lähestymistapoja probabilistisen päättelyn toteuttamiseksi monipolkuisissa Bayes-verkoissa on kuvattu lähteissä [18, 56, 31, 116, 85, 143].

3.3 Bayes-verkkojen rakentaminen

Kuten luvussa 3.1 näimme, Bayes-verkkomallien komponenteille voidaan antaa intuitiivisesti selkeä semanttinen tulkinta: Bayes-verkkostruktuuri kuvaa ongelmakentän kuvaamisessa käytettyjen attribuuttien välisiä riippuvuuksia, Bayes-verkon parametreina käytetyt ehdolliset todennäköisyydet taas riippuvuuksien voimakkuuksia. Tämäntyyppinen tieto on usein helposti saatavilla ongelmakentän asiantuntijoilta, jolloin Bayes-verkkojen konstruointi on helppoa. Bayes-verkkokehittimiä (katso luku 4.2) käyttäen Bayes-verkkojen soveltaminen ei edellytä syvempää Bayes-verkkoteorian ymmärtämistä, jolloin asiantuntijat voivat myös konstruoida Bayes-verkkomalleja suoraan itse, ja siirtyä samalla testaamaan kehitettyjen mallien toimivuutta. Eric Horvitzin mukaan[3] United Airlinesin insinöörit testasivat ensimmäisiä itse kehittämiään Bayes-verkkomalleja jo muutaman tunnin kuluttua siitä kun Bayes-verkkomallien käsite ja niiden konstruoimiseen soveltuvat kehittimet oli heille esitelty.

Toisaalta, kuten luvussa 1.2 todettiin, bayesiläisen mallinnuksen yksi suurimpia vahvuuksia on se, kuinka lähestymistapa mahdollistaa asiantuntijatietämyksen yhdistämisen tilastolliseen oppimiseen esimerkkiaineistosta. Mallien oppiminen jaettiin edellä kahteen vaiheeseen: mallistruktuurin oppiminen

[3] Katso URL: http://www.auai.org/BN-Testimonial.html

ja malliparametrien oppiminen. Kuvaamme seuraavassa kuinka laskea oppimiskriteeri eri Bayes-verkkostruktuureille, ja kuinka kiinnittää Bayes-verkon parametrit, annettuna struktuuri. Samassa yhteydessä selvitämme kuinka asiantuntijatietämys voidaan priorijakaumia käyttäen ottaa huomioon oppimisprosessissa.

3.3.1 Verkkostruktuurin oppiminen

Luvussa 1.2.2 näimme että bayesiläisessä oppimisessa mallistruktuurit evaluoidaan vertailemalla mallien posterioritodennäköisyyksiä

$$P(M \mid \mathcal{D}) \propto P(\mathcal{D} \mid M)P(M),$$

ja luvussa 1.2.4 näimme kuinka asiantuntijatietämys voidaan integroida oppimisprosessiin priorijakaumien kautta. Mallistruktuurin oppimisen päämääränä on siis iteratiivisen etsintäalgoritmin kautta etsiä mallistruktuuri M joka maksimoi tämän todennäköisyyden. Bayes-verkkojen tapauksessa etsintäavaruus on valtava: erilaisten syklittömien suuntamattomien verkkojen lukumäärä saadaan lähteessä [114] esitetystä rekursioyhtälöstä, josta näemme että jo kymmenellä solmulla saadaan aikaan noin 4.2×10^{18} erilaista Bayes-verkkostruktuuria! Tässä valossa ei olekaan lainkaan yllättävää että Bayes-verkkojen oppimisongelman on osoitettu olevan NP-kova ongelma [25]. On kuitenkin jälleen kerran korostettava, että kyseessä on pahimman tapauksen analyysi, joka ei kerro mitään ongelman keskimääräisestä vaikeudesta. Bayes-verkkojen oppimisessa onkin sovellettu hyvällä menestyksellä tunnettuja heuristisia etsintäalgoritmeja — viitteitä käytettyihin algoritmeihin löytyy esimerkiksi teoksista [48, 122].

Jos ongelmakentästä ei ole saatavilla priori-informaatiota, on järkevää olettaa eri struktuureille tasainen priorijakauma, jolloin todennäköisyyttä $P(M)$ voidaan pitää vakiona, ja käyttää oppimiskriteerinä yksinomaan kokonaisuskottavuutta $P(\mathcal{D} \mid M)$. Mikäli priori-informaatiota on saatavilla, voidaan tätä tietämystä käyttää hyväksi priorijakauman $P(M)$ määräämisessä. Koska mahdollisten Bayes-verkkojen määrä on kuitenkin tavattoman suuri, ei ole käytännössä mahdollista määrätä prioritodennäköisyyttä $P(M)$ erikseen kullekin Bayes-verkkostruktuurille. Priori-informaatiota voidaan kuitenkin käyttää hyväksi yhden (tai muutaman) ns. prioriverkon konstruoimisessa. Prioriverkoilla tarkoitetaan verkkostruktuureja joiden prioritodennäköisyden arvellaan olevan hyvin suuri (verrattuna muihin verkkostruktuureihin). Prioriverkkoja voidaan käyttää hyödyksi mallistruktuurin etsimisessä esimerkiksi heurististen etsintäalgoritmien aloituspisteinä, tai varsinaisen priorijakauman estimoinnissa lähteessä [50] esitetyllä tavalla.

Bayes-verkkojen kokonaisuskottavuuden $P(\mathcal{D} \mid M)$ laskemiseksi on tehtävä joitakin teknisiä oletuksia, joista tärkein koskee parametrien priorijakaumaa $P(\theta \mid M)$. Kuten luvussa 3.1 näimme, Bayes-verkkostruktuuriin liitettävät parametrit θ ovat muotoa $P(X \mid F(X))$ olevia ehdollisia todennäköisyyksiä, missä $F(X)$ on muuttujan X edeltäjien joukko. Numeroidaan joukon $F(X)$ kaikki mahdolliset arvokombinaatiot, ja oletetaan että niitä on q_i kappaletta. Merkinnällä θ_{ijk} tarkoitamme seuraavassa parametria $P(X_i = k \mid F(X_i) = j)$. Jos oletamme nyt että priorijakauma $P(\theta_{ij} \mid M)$ on ns. *Dirichlet-jakauma* (ks. esim. [39]) *hyperparametrein*[4] $N'_{ij1}, \ldots, N'_{ijr_i}$, niin voidaan osoittaa [28, 50] että kokonaisuskottavuus voidaan laskea kaavalla (3.4).

Kaavassa (3.4) esiintyvät tilastolliset tunnusluvut N_{ijk} on helppo laskea opetusjoukosta $\mathcal{D}$, ja kokonaisuskottavuus voidaan laskea ajassa $O(Nn^2r)$, missä N on joukon $\mathcal{D}$ koko, n muuttujien lukumäärä, ja r suurin mahdollinen muuttujien arvojen lukumäärä. Käyttäjän tehtäväksi jää määrätä hyperparametrit N'_{ijk}. Mikäli prioritietämystä ei ole, voidaan käyttää tasaista priorijakauma joka saadaan asettamalla $N'_{ijk} = 1$ kaikilla i:n j:n, ja k:n arvoilla. Jos prioritietämystä on saatavilla, antaa kaava (3.4) vihjeen siitä kuinka asiantuntijatietämys voidaan koodata hyperparametreiksi N'_{ijk}: hyperparametrit ja tilastolliset tunnusluvut N_{ijk} ovat käytettävässä kaavassa täysin identtisessä asemassa, joten on helppo päätellä, että jos käyttäjä asettaa hyperparametrin N'_{ijk} arvoksi esimerkiksi viisi, on tämän asetuksen vaikutus kokonaisuskottavuuteen sama kuin jos käyttäjä olisi lisännyt opetusjoukkoon $\mathcal{D}$ viisi esimerkkiä, joissa kussakin muuttujan X_i arvo olisi k, ja muuttujan edeltäjät $F(X_i)$ olisivat arvokombinaatiossa j. Käyttäjä voi siis muotoilla henkilökohtaisen näkemyksensä muuttujan X_i arvojen jakaumasta muuttamalla hyperparametreja N'_{ijk}, ja hyperparametrien määräämisessä käytetyn arvoalueen skaalaa muuttamalla saadaan vaihdeltua prioritietämyksen vaikutuksen suhdetta opetusjoukkoon vaikutukseen verrattuna. Tätä lähestymistapaa kutsutaan *ekvivalentin otoskoon (equivalent sample size)* menetelmäksi: asiantuntija voi painottaa tietämyksensä merkitystä ajattelemalla sitä "prioriopetusjoukkona", joukkona opetusjoukon ulkopuolisia tapauksia joista asiantuntija on oman kokemusperäisen tietämyksensä muodostanut.

Kaavassa (3.4) esitetty menetelmä kokonaisuskottavuuden laskemiseksi pätee diskreeteille muuttujille, kun muuttujien arvot otosjoukossa $\mathcal{D}$ oletetaan jakautuneiksi multinomijakauman mukaan. Vastaavanlainen tulos voidaan kuitenkin itse asiassa johtaa hyvin laajalle joukolle erilaisia jakaumia

[4]Nimitys "hyperparametri" johtuu siitä että kyseessä on "toisen kertaluvun" parametri, yhteistodennäköisyysjakauman määrittelemisessä käytettävän parametrin jakauman määrittelemisessä käytetty parametri.

Kokonaisuskottavuuden laskeminen:

$$P(\mathcal{D} \mid M) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(N'_{ij})}{\Gamma(N'_{ij} + N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(N'_{ijk} + N_{ijk})}{\Gamma(N'_{ijk})}, \tag{3.4}$$

missä

- Γ on ns. Gamma-funktio, tavallisen kertomafunktion yleistys myös muille kuin kokonaisluvuille (minkä ansiosta hyperparametrien ei tarvitse välttämättä olla kokonaislukuarvoisia),
- n on muuttujien lukumäärä,
- q_i on Solmun X_i edeltäjien mahdollisten arvokombinaatioiden lukumäärä.
- r_i on muuttujan X_i arvojen lukumäärä,
- N_{ijk} on niiden tapausten lukumäärä joukossa $\mathcal{D}$, missä muuttujan X_i arvo on k, ja muuttujan X_i edeltäjät $F(X_i)$ ovat arvokombinaatiossa j,
- N'_{ijk} on parametria θ_{ijk} vastaavan priorijakauman määräämisessä käytetty hyperparametri,
- $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$, ja
- $N'_{ij} = \sum_{k=1}^{r_i} N'_{ijk}$.

(ns. eksponentiaaliselle perheelle), myös jatkuville jakaumille, ja jatkuvien ja diskreettien muuttujien muodostamille sekajakaumille [10]. Esimerkki Bayes-verkon kokonaisuskottavuuden laskemisesta jatkuvien jakaumien tapauksessa löytyy lähteestä [40].

Yllä kuvattu lähestymistapa perustuu Bayes-verkkostruktuurien vertailuun niiden posterioritodennäköisyyden (tai käytännössä yleensä kokonaisuskottavuuden) perusteella. Luvussa 3.1 esitetty Bayes-verkkojen määritelmä tarjoaa myös vaihtoehtoisen lähestymistavan tähän ongelmaan: koska kukin Bayes-verkkostruktuuri vastaa tiettyä joukkoa riippumattomuusoletuksia, voidaan opetusjoukosta ensin tilastollisesti päätellä ongelmakentässä vallitsevat riippumattomuudet, ja konstruoida sitten pääteltyjä riippumattomuuksia vastaava Bayes-verkkostruktuuri. Tähän lähestymistapaan perustuvia algoritmeja on esitetty esimerkiksi lähteessä [130]. Täydellisten riippumattomuuksien sijaan voidaan yrittää tilastollisesti päätellä muuttujien osittaisia järjestyksiä siten, että syntyvässä Bayes-verkossa muuttujasta X voi olla kaari muuttujaan Y vain jos X on Y:n edeltäjä kiinnitetyssä järjestyksessä. Tilastollisesti pääteltyjä osittaisia järjestyksiä voidaan myös yhdistää yllä esitettyyn MAP-metriikkaan perustuviin oppimisalgoritmeihin [125]. Osittaisen järjestyksen määräämiseksi voidaan tilastollisten testien sijaan käyttää tietenkin myös priori-informaatiota. Itse asiassa monet MAP-metriikkaa soveltavat oppimisalgoritmit, kuten esimerkiksi Cooperin ja Herskovitsin K2-algoritmi [28], edellyttävät tällaisen osittaisen järjestyksen määräämistä ennen oppimisprosessin alkua.

3.3.2 Bayes-verkon parametrien oppiminen

Suurimman todennäköisyyden parametrit Kun Bayes-verkkostruktuuri M on kiinnitetty, on seuraava vaihe mallin oppimisessa mallin parametrien θ valinta. Kuten luvussa 1.2.3 näimme, bayesiläisessä lähestymistavassa parametrit kiinnitetään MAP-arvoonsa, siis arvoon $\hat{\theta}$ joka maksimoi posterioritodennäköisyyden $P(\theta \mid \mathcal{D}, M)$:

$$\hat{\theta} = \arg\max_{\theta} P(\theta \mid \mathcal{D}, M).$$

Samojen teknisten oletusten vallitessa, joita käytettiin edellisessä luvussa esitetyn kokonaisuskottavuuden kaavan (3.4) johtamisessa, voidaan osoittaa että malliparametrien posterioritodennäköisyys maksimoituu asettamalla

$$\hat{\theta}_{ijk} = \frac{N_{ijk} + N'_{ijk} - 1}{N_{ij} + N'_{ij} - r_i}, \tag{3.5}$$

missä merkinnät ovat kuten edellä kaavassa (3.4). MAP-parametriarvojen laskemisessa ei siis tarvita iteratiivista oppimisalgoritmia, vaan ne voidaan määrätä suoraan käyttäen kaavaa (3.5)!

Suurimman uskottavuuden parametrit On tärkeää huomata, että yllä annettu laskukaava MAP-parametrien määräämiseksi poikkeaa klassisesta tilastotieteestä tutusta *suurimman uskottavuuden (maximum likelihood, ML)* parametriasetuksesta $\tilde{\theta}$,

$$\tilde{\theta}_{ijk} = \frac{N_{ijk}}{N_{ij}} = \frac{N_{ijk}}{\sum_{k=1}^{r_i} N_{ijk}}, \tag{3.6}$$

joka maksimoi datan uskottavuuden $P(\mathcal{D} \mid \theta, M)$. Koska

$$P(\theta \mid \mathcal{D}, M) \propto P(\theta \mid M) P(\mathcal{D} \mid \theta, M),$$

näemme että ML- ja MAP-parametrit ovat samat siinä tapauksessa, että priorijakauma $P(\theta \mid M)$ on tasainen, ja siis vakio. Suurimman uskottavuuden parametrit saadaan siis erikoistapauksena kaavasta (3.5), jos käytämme tasaisia priorijakaumia, eli asetamme kaikkien hyperparametrien N'_{ijk} arvoksi yksi.

Odotusarvoparametrit Parametrien θ asettaminen MAP-arvoihinsa on bayesiläisen perusidelogian mukainen vastaus kysymykseen kuinka kiinnittää parametrit annettuna mallistruktuuri. Mikäli syntyvää mallia (M, θ) kuitenkin käytetään probabilistisessa päättelyssä (eikä siis ainoastaan ongelmakentän analysoinnissa), näemme hiukan yllättäen että MAP-parametreja käyttävälle mallille löytyy vaihtoehto, joka tuottaa tarkempia ennustuksia probabilistisessä päättelyssä. Olkoon nimittäin $P(X_1, \ldots, Xn \mid \theta, M)$ mallin (M, θ) tuottama jakauma. Koska emme tiedä varmasti mikä mallin θ tuottama jakauma vastaa ongelmakentän todellista todennäköisyysjakaumaa, arvioimme eri mallien hyvyyttä niiden posterioritodennäköisyyden $P(\theta \mid \mathcal{D}, M)$ mukaan käyttämällä apuna ongelmakentästä saatua otosta $\mathcal{D}$. Yllä esitetyssä MAP-lähestymistavassa valittiin siis parametrit, jotka maksimoivat tämän todennäköisyyden. Tarkempi jakauma saataisiin kuitenkin aikaan ottamalla esimerkiksi kaksi tässä mielessä parasta mallia θ_1 ja θ_2, ja muodostamalla niiden painotettu keskiarvo:

$$P(X_1, \ldots, Xn \mid \theta_1, M) P(\theta_1 \mid \mathcal{D}, M) + P(X_1, \ldots, Xn \mid \theta_2, M) P(\theta_2 \mid \mathcal{D}, M).$$

Seuraamalla tätä päättelyketjua eteenpäin, näemme että optimaalisessa tapauksessa meidän pitäisi käyttää *kaikkia (äärettömän monia) mahdollisia parametriasetuksia*, jolloin saamme jakauman

$$\int P(X_1, \ldots, Xn \mid \theta, M) P(\theta \mid \mathcal{D}, M) d\theta = P(X_1, \ldots, Xn \mid M).$$

Voidaan kuitenkin osoittaa että syntynyt jakauma on identtinen jakauman kanssa, joka saadaan käyttäen yhtä mallia jossa kukin parametriarvo on kiinnitetty odotusarvoonsa $E(\theta_{ijk} \mid \mathcal{D}, M)$. Odotusarvomallin käyttäminen vastaa siis, paradoksaalista kyllä, tilannetta jossa emme kiinnitä parametriarvoa lainkaan vaan käytämme kaikkia äärettömän monia parametriasetuksia! Odotusarvot saadaan laskettua kaavasta

$$\hat{\theta}_{ijk} = E(\theta_{ijk} \mid \mathcal{D}, M) = \frac{N_{ijk} + N'_{ijk}}{N_{ij} + N'_{ij}}. \tag{3.7}$$

Odotusarvoihin perustuvien mallien on empiirisesti havaittu tuottavan paljon tarkempia estimaatteja ongelmakentän todennäköisyysjakaumasta, etenkin kun käytettävissä olevan opetusjoukon $\mathcal{D}$ koko on pieni [74, 75, 133].

Kolme laskukaavaa malliparametrien määräämiseksi:

- Suurimman todennäköisyyden parametrit (3.5):

$$\theta_{ijk} = \frac{N_{ijk} + N'_{ijk} - 1}{N_{ij} + N'_{ij} - r_i}.$$

- Suurimman uskottavuuden parametrit (3.6):

$$\theta_{ijk} = \frac{N_{ijk}}{N_{ij}}.$$

- Odotusarvoparametrit (3.7):

$$\theta_{ijk} = \frac{N_{ijk} + N'_{ijk}}{N_{ij} + N'_{ij}}.$$

3.4 Bayes-verkkojen variaatioita

3.4.1 Päätösverkot (influence diagrams)

Luvussa 1.4 näimme kuinka probabilistisessa päätelyssä tuotettuja ehdollisia todennäköisyysjakaumia voidaan käyttää hyväksi päätöksentekoprosessissa päätösteoriana tunnetun formalismin kautta. *Päätösverkko (influence diagrams)* on Bayes-verkko, jossa päätöksentekoon liittyvät vaihtoehdot koodataan erityisinä päätössolmuina, ja päätössolmujen eri arvoihin liittyvä odotettavissa olevan hyödyn määrä lasketaan suoraan probabilistisen päättelyn yhteydessä (katso esim. [123, 101]). Päätössolmujen käsittelymahdollisuus on liitetty mukaan useisiin Bayes-verkkoja manipuloiviin ohjelmistoihin (katso luku 4.2).

3.4.2 "Noisy-or"-malli

Kuten luvussa 3.2 näimme, probabilistisen päättelyn eri muodot voidaan toteuttaa yksipolkuisilla (joko aidosti yksipolkuisilla, tai monipolkuista verkkoa vastaavalla liittymäpuulla) Bayes-verkoilla ajassa joka on suoraan verrannollinen verkkoa vastaavien parametrien määrään. Luvussa 3.1 näimme että nämä parametrit ovat muotoa $P(X \mid F(X))$, missä $F(X)$ on solmun X edeltäjien joukko annetussa verkossa. Tätä muotoa olevia parametreja on eksponentiaalinen määrä solmun X edeltäjien arvojen lukumäärän suhteen, joten jos solmulla on hyvin suuri joukko edeltäjiä, saattaa tarvittavien parametrien lukumäärä olla liian suuri käytännön tarpeita ajatellen[5]. Ns. “noisy-or”-mallissa muotoa $P(X \mid F(X))$ olevia parametreja approksimoidaan kaavan

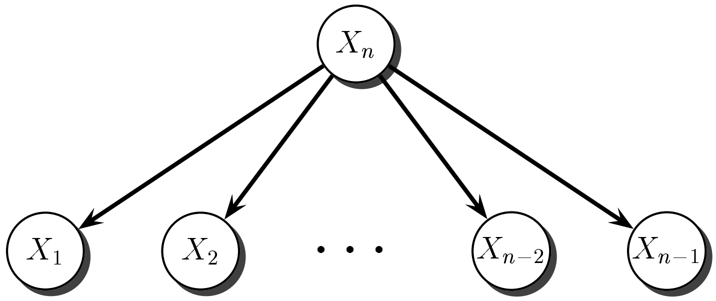
$$P(X \mid Y_1, \ldots, Y_n) = 1 - \prod_{i=1}^{n}(1 - P(X \mid Y_i))$$

mukaan. “Noisy-or”-mallia käyttäen voidaan tarvittavien parametrien lukumäärää pienentää siis huomattavasti: kutakin solmua X kohden riittää tallettaa $|F(X)|$ parametria, missä $|F(X)|$ on solmun edeltäjien lukumäärä. “Noisy-or”-approksimaation on empiirisesti havaittu toimivan useissa käytännön tilanteissa hyvin, vaikka toisinaan sen sisältämät implisiittiset riippumattomuusoletukset eivät pidä paikkaansa jolloin approksimaatio ei ole riittävän tarkka käytännön sovelluksia ajatellen [53, 97].

3.4.3 Jatkuva-arvoiset muuttujat

Tässä raportissa on havainnollisuuden vuoksi keskitytty tarkastelemaan diskreettejä Bayes-verkkoja, joissa erilaisia mahdollisia muuttujien arvoja on vain äärellinen määrä. Koska Bayes-verkko kuitenkin viime kädessä on yhteistodennäköisyysjakauman hajotelman graafinen esitys, joka kertoo kuinka muuttujajoukon yhteistodennäköisyys saadaan laskettua ehdollisten todennäköisyyksien tulona (katso kaava 3.3), ei mikään estä käyttämästä tässä hajotelmassa myös jatkuvia ehdollisia tiheysfunktioita: sekä luvussa 3.2 hahmoteltu probabilistisen päättelyn toteuttamiseen käytetty viestinvälitysalgoritmi [83, 3], että luvussa esitetty Bayes-verkkojen rakentamisessa käytetty mallinvalintakriteeri [40] voidaan määritellä myös jatkuvien muuttujien tapauksessa, ja myös sekatapauksessa jossa käytetään sekä jatkuvia että diskreettejä muuttujia. Malliperheen laajennus tällä tavalla saattaa joissakin tapauksissa aiheuttaa teknisiä ongelmia — yksinkertaisinta jatkuvien muuttujien käsittely on ns. äärellisissä sekajakaumamalleissa, joita käsittelemme luvussa 3.4.5. On myös huomattava että käytännön sovelluksissa reaaliarvoisten muuttujienkaan arvot eivät todellisuudessa ole äärettömän tarkkuuden reaalilukuja, vaan jollakin äärellisellä tarkkuudella esitettyjä numeroita. Niinpä monissa sovelluksissa jatkuvia arvoja voidaankin yhtä hyvin käsitellä suoraan diskreetteinä muuttujina, tai ne voidaan ensin diskretisoida. Bayesiläistä lähestymistapaa diskretisointiin on käsitelty lähteissä [37, 71].

[5]Useimmat käytännön sovelluksissa käytettävät Bayes-verkot ovat suhteellsien harvoja, jolloin tätä ongelmaa ei esiinny.



Kuva 3.8: Naiivi Bayes-mallin Bayes-verkkoesitys. Juurimuuttuja X_n on luokkamuuttuja, jonka suhteen muiden muuttujien riippumattomuusoletus tehdään.

3.4.4 Naiivi Bayes-malli

Luokitteluongelmissa tehtävänä on laskea diskreetin luokkamuuttujan jakauma annettuna muiden muuttujien (mahdollisesti osittainen) arvokombinaatio $\mathbf{S}_2 = \mathbf{s}_2$. Niin sanotussa *Naiivi Bayes*-mallissa muut muuttujat oletetaan riippumattomiksi, jos luokkamuuttujan arvo on tunnettu. Tästä oletuksesta seuraa että muuttujien yhteistodennäköisyysjakauma voidaan esittää muodossa

$$P(X_1, \ldots, X_n) = P(X_1, \ldots, X_{n-1} \mid X_n)P(X_n) = P(X_n) \prod_{i=1}^{n-1} P(X_i \mid X_n),$$

missä muuttujat on oletettu järjestetyiksi siten että luokkamuuttujan indeksi on n. Tämän hajotelman Bayes-verkkoesitys on kuvassa 3.8.

Naiivi Bayes-mallia on sovellettu monissa erilaisissa luokitteluongelmissa varsin hyvällä menestyksellä. Yksi syy mallin suosioon on probabilistisen päättelyn tehokkuus: koska Bayes-verkko on tässä tapauksessa kaksitasoinen puu, on laskenta äärimmäisen tehokasta. Itse asiassa Naiivi Bayes-mallissa tapahtuva probabilistinen päättely voidaan toteuttaa feedforward-neuroverkon tapaisena arkkitehtuurina [95, 94]. Mallin toinen merkittävä etu on se että koska mallin arkkitehtuuri on kiinnitetty, ei mallistruktuurin oppimisongelmaa ole, ja mallin parametrit voidaan asettaa suoraan luvussa 3.3.2 esitettyjä kaavoja käyttäen.

Vaikka Naiivi Bayes-malliin johtava riippumattomuusoletus saattaa tuntua melko epärealistiselta, toimii Naiivi Bayes-malli käytännössä usein yllättävän hyvin [133, 82]. Itse asiassa monet Bayes-verkkojen kaupallisista sovelluksista käyttävät tätä yksinkertaista mallia (esimerkiksi Microsoftin Answer Wizard ja Office Assistant, ks. luku 4.1). Hyvä esimerkki Naiivi Bayes-mallin sovellettavuudesta saatiin “Knowledge Discovery and Data Mining”-konferenssin yhteydessä kesällä 1997 julkistetussa maailmanlaajuisessa ennustuskilpailussa, jossa kaksi kolmesta parhaasta ratkaisusta käytti juuri Naiivi Bayes-mallia [103]. Tämän yksinkertaisen mallin menestykselle voidaan antaa seuraava intuitiivinen selitys: monissa luokitteluongelmissa (esimerkiksi lääketieteellisissä ongelmissa) on muuttujien arvojen määrittäminen kallista, koska se edellyttää esimerkiksi erityisten laboratoriokokeiden suorittamista. Niinpä kahden toisistaan riippuvan muuttujan mukaanottaminen on resurssien tuhlausta, koska jo toinenkin näistä muuttujista sisältää tarvittavan informaation. Vaikuttaa siltä että monet reaalimaailman datajoukot on kerätty resursseja säästäen, minkä seurauksena Naiivi Bayes-mallin riippumattomuusoletus saattaa olla hyvinkin järkevä.

Naiivi Bayes-mallin on toisaalta havaittu luokittelevan hyvin myös sellaisissa (keinotekoisissa) tilanteissa, joissa käytetty datajoukko on konstruoitu sellaiseksi, että Naiivi Bayes-mallin riippumattomuusoletus rikkoutuu varmasti [5]. Mallin luokittelutarkkuus on näissäkin tilanteissa ollut suuri — malli pystyy siis diskriminoimaan suurella tarkkuudella annetut esimerkit vastaaviin luokkiin — mutta jos tarkastellaan mallin antamaa jakaumaa luokkamuuttujalle, nähdään että mallin estimoima jakauma ei ole kovinkaan tarkka. Naiivi Bayes-mallin riippumattomuusoletuksesta seuraa siis ilmeisen konsistetti vinouma (bias) estimoituun luokkamuuttujan jakaumaan. Vinouma estää luokkamuuttujan jakauman tarkan estimoinnin, mutta koska oletuksista seurannut vinouma on kaikissa tilanteissa johdonmukainen, saattaa menetelmän erottelukyky olla edelleen hyvä. Tämä tarkoittaa siis sitä että Naiivi Bayes-malli saattaa olla hyvä vaihtoehto silloin jos luokittelussa riittää erotella annetut tapaukset eri luokkiin, mutta naiivi Bayes-mallin tarjoaman luokkamuuttujan jakauman käyttöön riskianalyysissä kannattaa suhtautua varauksella.

3.4.5 Latentit muuttujat ja äärelliset sekajakaumat

Standardit Bayes-verkkomallit perustuvat siihen oletukseen että mallintamisessa käytettävät muuttujat $X_1, \ldots, X_n$ on valittu siten että ongelmakentälle voidaan konstruoida toimiva malli valittuja muuttujia käyttäen. Mikäli näin ei ole, voidaan tietysti kyseenalaistaa koko tehtävänasettelu, ja valita käytettävät muuttujat uudelleen. Jos muuttujajoukon uudelleenmäärittely ei kuitenkaan tällaisessa tilanteessa ole mahdollista, saattaa Bayes-verkkojen (tai minkä tahansa vain annettuja muuttujia käyttävän malliperheen) soveltaminen tuottaa epätyydyttäviä tuloksia. Seuraava yksinkertainen esimerkki havainnollistaa tätä seikkaa.

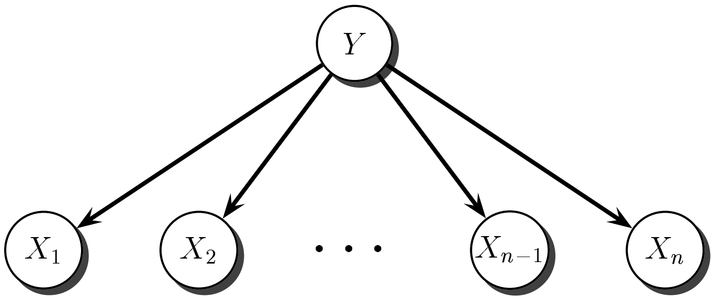
Oletetaan että käytettävissä olevia muuttujia on kaksi, X_1, ja X_2, ja muuttujan X_1 arvo kertoo hukkumiskuoleminen viikottaisen lukumäärän Suomessa, ja muuttujan X_2 arvo Suomessa viikottain myydyn jäätelön määrän. Tilastojen perusteella on helppo nähdä että muuttujilla on selvästi jonkinlainen riippuvuussuhde, joten Bayes-verkkostruktuuri jossa muuttujat eivät ole yhdistettyjä toisiinsa, ei mallinna ongelmakenttää tyydyttävällä tavalla. Jäljelle jää vain kaksi mahdollista Bayes-verkkostruktuuria: $X_1 \rightarrow X_2$ ja $X_1 \leftarrow X_2$. Intuitiivisesti ottaen nämä tapaukset näyttäisivät kuitenkin indikoivan että jompi kumpi muuttuja olisi toisen muuttujan syy, mikä ei varmaankaan pidä paikkaansa todellisuudessa. Jos meidän nyt sallitaan lisätä malliimme uusi muuttuja Y, niin näemme että Bayes-verkkorakenne $X_1 \leftarrow Y \rightarrow X_2$ tuottaa muuttujien X_1 ja X_2 välille *epäsuoran riippuvuussuhteen*, joka mallintaa tyydyttävästi intuitiivista kuvaamme ongelmakentästä.

Edellisessä esimerkissä käytettyä muuttujaa Y kutsutaan *latentiksi muuttujaksi* tai *piilomuuttujaksi*, koska muuttuja ei ollut mukana alkuperäisessä muuttujajoukossa, ja sen arvoja ei siten ole annettu opetusjoukossa $\mathcal{D}$. Yksinkertaisessa esimerkissämme on käytetylle latentille muuttujalle helppo keksiä semanttinen tulkinta (vuodenaika), mutta aina näin ei ole asianlaita, eikä se bayesiläisen mallintamisen kannalta ole välttämättä tarpeenkaan: latenttien muuttujien arvoja opetusjoukossa voidaan pitää puuttuvana datana, joka marginalisoidaan pois käytetyistä laskukaavoista.

Mikäli latentteja muuttujia on vain yksi, ja latentista muuttujasta on kaari kaikkiin muihin muuttujiin, ja muiden muuttujien välillä ei ole kaaria, kutsutaan syntyvää Bayes-verkkomallia *äärelliseksi sekajakaumamalliksi (finite mixture model)* [35, 136]. Syntyvä yhteistodennäköisyysjakauma on tässä tapauksessa

$$\begin{aligned} P(X_1,\ldots,X_n) &= \sum_y P(X_1,\ldots,X_n,Y=y) \\ &= \sum_y P(X_1,\ldots,X_n \mid Y=y)P(Y=y) \\ &= \sum_y P(Y=y)\prod_{i=1}^{n} P(X_i \mid Y=y), \end{aligned}$$

missä summa käy yli kaikkien mahdollisten piilomuuttujan Y arvojen. Vastaava Bayes-verkkostruktuurin graafinen esitys on kuvassa 3.9. On tärkeää huomata että vaikka syntyvä mallistruktuuri on muodollisesti identtinen Naiivi Bayes-mallin (kuva 3.8) kanssa, on juurimuuttuja sekajaumamallien tapauksessa latentti muuttuja (ja mahdollinen luokkamuuttuja on siis puun lehti siinä missä muutkin muuttujat), joten malli ei tee samoja riippumattomuusoletuksia kuin Naiivi Bayes-malli. Itse asiassa on helppo nähdä että



Kuva 3.9: Äärellisen sekajakaumamallin Bayes-verkkoesitys. Juurimuuttuja Y on latentti piilomuuttuja, jonka suhteen muiden muuttujien riippumattomuusoletus tehdään.

mikä tahansa Bayes-verkkomalli voidaan muuttaa äärelliseksi sekajakaumamalliksi, eli sekajakaumamallit muodostavat universaalin malliperheen.

Bayes-verkkoihin verrattuna sekajakaumamalleilla on se etu, että koska syntyvä Bayes-verkkostruktuuri on yksinkertainen puu, on sekajakaumamalleilla suoritettava päättely laskennallisesti aina erittäin tehokasta. Lisäksi mahdollisia verkkostruktuureja on vain yksi, joten mallistruktuuriavaruus ei ole kovin suuri: ainoa vapaa parametri on latentin muuttujan arvojen lukumäärä, jonka voidaan olettaa olevan vähemmän kuin joukon $\mathcal{D}$ koko. Juurimuuttujan latenttiudesta seuraa kuitenkin että kokonaisuskottavuus $P(\mathcal{D} \mid M)$, joka Bayes-verkkojen tapauksessa lasketaan helposti kaavaa (3.4) käyttäen, ei ole sekajakaumien tapauksessa laskettavissa käytännössä tarkasti: kokonaisuskottavuus saadaan marginalisoimalla yli kaikkien mahdollisten piilomuuttujan arvoista muodostuvien vektorien Z yli,

$$P(\mathcal{D} \mid M) = \sum_{Z} P(\mathcal{D}, Z \mid M),$$

joita on eksponentiaalinen määrä. Sekajakaumamallien kokonaisuskottavuuden laskemiseksi on kuitenkin kehitetty monia tehokkaita approksimaatiomenetelmiä (katso [73, 81, 77, 72, 26]).

Kokonaisuskottavuuden laskemisen lisäksi piilomuuttujaoletuksesta implisiittisesti seuraava puuttuva informaatio tekee myös sekajakaumamallien malliparametrien laskemisen vaikeammaksi kuin normaalien Bayes-verkkojen tapauksessa. Tähänkin tehtävään on kehitetty tehokkaita approksimaatiomenetelmiä, joista yleisin tunnetaan nimella *Expectation-Maximization (EM)* [33, 91, 62, 13, 78]. Koska latenttia piilomuuttujaa Y voidaan ajatella eräänlaisena "klusterimuuttujana", joka ryhmittelee annetut esimerkkitapaukset, voidaan EM-algoritmia pitää probabilistisena ryhmittelyalgoritmina. Algoritmin tuottamia tapausryhmiä voidaan soveltaa ongelmakentän probabilistisessa analysoinnissa ja visualisoinnissa (data mining) samalla tavoin kuin esimerkiksi SOM-neuroverkkomallia [67]. Algoritmi tarjoaa siis bayesiläisen lähestymistavan *numeerisena taksonomiana* tunnetulle tutkimusalueelle [43].

Kuten luvussa 2.3 todettiin, sekajakaumamalliperhe tarjoaa yhtenäisen teoreettisen kehikon monille näennäisesti erilaisille lähestymistavoille, kuten esimerkkeihin perustuva päättely [135, 134, 95, 69, 70], ydinestimaattorit, ydinkantafunktiot (radial basis functions), sekä probabilistiset neuroverkot [12], itseorganisoivat kartat [14], sekä piilo-Markov-mallit (hidden Markov models) [126, 30, 108]. Lähteissä [133, 135] raportoidut laajat empiiriset testitulokset osoittavat että mallistruktuurin näennäisestä yksinkertaisuudesta huolimatta sekajakaumallit (ja myös jopa Naiivi Bayes-mallit) tuottavat konsistentisti erittäin hyviä tuloksia vaihtoehtoisilla menetelmillä (kuten esimerkiksi neuroverkot ja päätöspuut) saavutettuihin tuloksiin verrattuna. Erityisen kiinnostavaa on että bayesiläiset mallit näyttävät suoriutuvan hyvin myös tilanteissa, joissa opetusdataa on hyvin vähän: saatavilla olevan datan määrän vähentäminen jopa 90 prosentilla ei näytä juurikaan huonontavan mallien antamia tuloksia [74, 75].

3.4.6 Kvalitatiiviset Bayes-verkot

Kvalitatiivisissa Bayes-verkoissa (qualitative Bayesian networks) [140, 34] luovutaan verkon parametrien (ehdollisten todennäköisyyksien) esittämisestä kvantitatiivisina reaalilukuina, ja siirrytään äärelliseen joukkoon mahdollisia kvalitatiivisia arvoja, esimerkiksi joukkoon $\{-, 0, +\}$. Verkon kaaret kuvaavat tässä tilanteessa kuinka tietyn solmun edeltäjät vaikuttavat vastaavaan muuttujaan X: vähentääkö ('-') vai lisääkö ('+') edeltäjien arvokombinaatio muuttujan X todennäköisyyttä, vai onko arvokombinaatio X:n suhteen irrelevantti ('0'). Näyttää siltä että kvalitatiivisten Bayes-verkkojen tapauksessa on mahdollista suorittaa määrättyjä probabilistisen päättelyn muotoja hyvin tehokkaasti myös monipolkuisten verkkojen tapauksessa, mutta teoreettisia tuloksia asiasta ei vielä juuri ole. Suomenkielinen katsaus kvalitatiivisia verkkoja käsittelevän tutkimuksen nykytilaan löytyy lähteestä [127].

3.4.7 Bayes-verkkojen ja neuroverkkojen yhteyksiä

Kuten edellä on jo mainittu, bayesiläisiä tekniikoita voidaan periaatteessa soveltaa myös muiden malliperheiden kuin Bayes-verkkojen tapauksessa. Mackay [88] ja Neal [96] kuvaavat väitöskirjoissaan bayesiläisten tekniikoiden soveltamista suunnattujen neuroverkkojen oppimisessa. Mackay voitti kehittämällään ohjelmistolla vuonna 1993 järjestetyn laajan aikasarjaennustamiskilpailun [89]. Sekä Mackayn[6] että Nealin[7] kehittämät ohjelmistot ovat vapaasti saatavilla kehittäjiensä kotisivuilta.

Paitsi neuroverkkojen oppimisalgoritmien kehittämisessä, Bayes-verkkojen ja neuroverkkojen välille on luotu myös lukuisia muita yhteyksiä. Michael Jordanin tutkimusryhmä MIT:ssa on tutkinut Bayes-verkkojen ja suunnattujen neuroverkkojen välimuotoa, jossa Bayes-verkkojen muuttujien välisten riippuvuuksien voimakkuuksia kuvataan neuroverkoista tutulla sigmoidfunktiolla. Malliperheestä käytetään nimitystä *sigmoid belief networks* [117]. Bishop kuvaa kirjassaan [13] ydinkantafunktioiden (radial basis functions), probabilististen neuroverkkojen, ydinestimaattoreiden ja sekajakaumallien välisiä yhteyksiä. Myllymäki kuvaa väitöskirjassaan [93] kuinka Bayes-verkoissa suoritettava probabilistinen päättely voidaan suorittaa massiivisesti rinnakkaisella Boltzmannin kone-neuroverkkoarkkitehtuurilla. Bayesiläinen kehikko itseorganisoiville kartoille [67] on kuvattu artikkelissa [14].

[6] URL: http://wol.ra.phy.cam.ac.uk/mackay/homepage.html
[7] URL: http://www.cs.utoronto.ca/~radford/

Luku 4

Bayes-verkkojen sovelluksia

Bayes-verkkojen sovellusesimerkkejä: lääketieteellinen diagnosointi, prosessikontrolli, vikadiagnostiikka, tiedon analysointi. Millä ohjelmistoilla Bayes-verkkomalleja voidaan rakentaa (Bayes-verkkokehittimet): kaupalliset ohjelmistot, tutkimusprototyypit.

Bays-verkkosovellukset voidaan jakaa kahteen ryhmään: yksittäiset Bayes-verkkotekniikkoihin perustuvat sovellukset, ja yleiset Bayes-verkkokehittimet, jotka ovat geneerisiä työkaluja erillisten sovellusten tuottamista varten. Bayes-verkkotutkimukseen on koko sen vasta noin kymmenvuotisen historian aikana liittynyt selkeä pyrkimys kehitettyjen menetelmien kaupalliseen soveltamiseen, ja sovellusten määrä onkin viime vuosina ollut voimakkaassa kasvussa. Lisäkiinnostusta Bayes-verkkojen hyödyntämismahdollisuuksia kohtaan herätti Bill Gatesin lokakuussa 1996 antama lausunto, jonka mukaan Microsoftin kilpailuetu tulevaisuudessa perustuu heidän eritysosaamiseensa Bayes-verkkojen alueella [54]. Microsoftilla onkin yksi maailman johtavia Bayes-verkkojen tutkimusyksiköitä (katso luku 5.2.1).

Luvussa 4.1 esitellään joitakin yksittäisiä Bayes-verkkosovelluksia. Luvussa ei pyritä antamaan tyhjentävää luetteloa kaikista olemassaolevista Bayes-verkkosovelluksista, vaan tyypillisiä esimerkkejä Bayes-verkkotekniikoita käyttävistä sovelluksista erilaisilta sovellusalueilta. Jotkut mainituista sovelluksista on luotu käyttäen kaupallisia Bayes-verkkokehittimiä, joita käsitellään luvussa 4.2, toiset taas ovat käsityönä luotuja räätälöityjä ohjelmistoja. Lisätietoa Bayes-verkkosovelluksista löytyy mm. David Heckermanin laatimasta listasta[1], johon on koottu yli sata viitettä Bayes-verkkojen käytännön sovelluksia raportoiviin artikkeleihin vuosilta 1985–1995. Heckermanin lista on melko laaja, mutta jo hieman vanhentunut. AUAI-järjestö (Association for

[1] URL: ftp://ftp.research.microsoft.com/pub/dtg/david/BN-APPS.TXT

Uncertainty in Artificial Intelligence) on koonnut suppeahkon listan joistakin merkittävimmistä viimeaikaisista sovelluksista. Lista on saatavilla järjeston kotisivulta[2], mistä löytyy myös kokoelma käytännön soveltajilta kerättyjä kommentteja siitä, miksi Bayes-verkot ovat osoittautuneet onnistuneeksi ratkaisuksi heidän sovellusympäristössään.

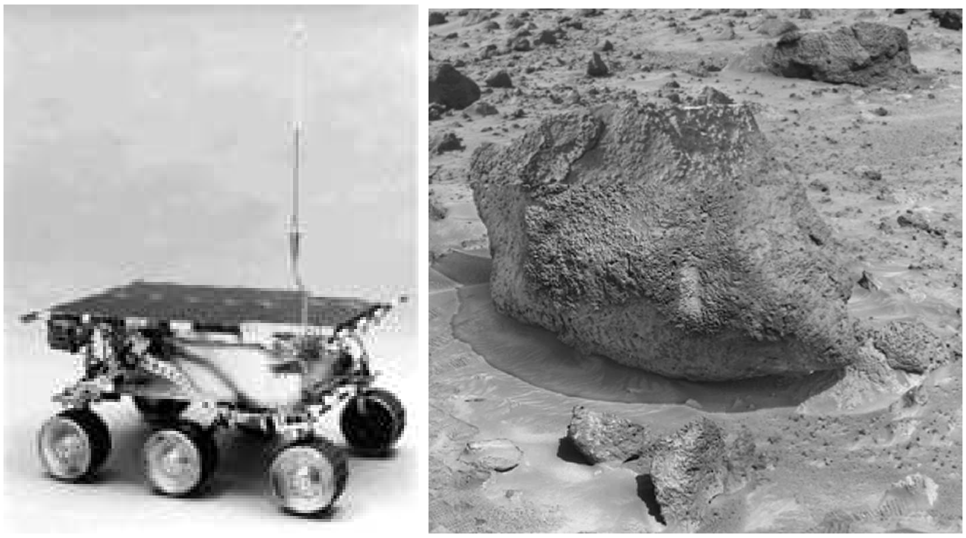
4.1 Sovellusesimerkkejä

Luvussa 3.2 esitettiin kuinka probabilistisen päättelyn eri muotoja voidaan toteuttaa käytännössä Bayes-verkkomalleja hyväksikäyttäen. Laajat julkisesti saatavilla olevilla luokitteluaineistoilla tehdyt vertailut osoittavat, että bayesiläiset mallit tuottavat konsistentisti erittäin hyviä tuloksia vaihtoehtoisilla menetelmillä (kuten esimerkiksi neuroverkot ja päätöspuut) saavutettuihin tuloksiin verrattuna [133, 135, 74, 79, 80]. Kuten luvussa 1.3 näimme, probabilistista päättelyä voidaan soveltaa luokitteluongelmien lisäksi ratkaisemaan hyvin monentyyppisiä ongelmatilanteita, joten Bayes-verkkojen sovellusalueitakin on lukuisia. Seuraavassa joitakin tyypillisimpiä sovellusesimerkkejä sovellusalueittain luokiteltuna.

Lääketieteellinen diagnosointi Lääketieteellinen diagnosointi on perinteisesti ollut yksi probabilististen menetelmien suosituimmista sovellusalueista. Yksi syy tähän varmaankin on se, että probabilististen menetelmien kanssa on tulosten luotettavuutta mahdollista arvioida tekemällä riskianalyysi luvussa 1.4 esitetyllä tavalla. Yksi ensimmäisiä tällaisia ohjelmistoja oli rintasyöpäpotilaiden diagnosointiin suunniteltu PathFinder [51], joka on myöhemmin kaupallistettu BiopSys Medical-yhtiön[3] IntelliPath-nimisenä tuotteena. IntelliPath-ohjelmisto on tällä hetkellä käytössä sadoissa eri sairaaloissa ympäri maailmaa. Muita vastaavia eri sairaaloissa ja tutkimuslaitoksissa aktiivisessa käytössä olleita järjestelmiä ovat mm. ACORN, DXPLAIN ja Iliad[4]. Yksi suurisuuntaisimpia tämän ongelma-alueen hankkeita on Microsoftin OnParenting-ohjelmisto[5], joka on lasten sairauksien diagnosointiin suunniteltu laaja järjestelmä. Järjestelmä käyttää sairauksien diagnosointiin Knowledge Industries-yhtiön kehittämää Bayes-verkkomallia.

Tiedon analysointi (data mining) Koska Bayes-verkkoja käyttäen voidaan tilastollisesta esimerkkiaineistosta konstruoida ongelmakentästä semant-

[2] URL: http://www.auai.org/
[3] URL: http://www.biopsys.cim
[4] URL: http://www-uk.hpl.hp.com/people/ewc/list-main.html
[5] URL: http://www.onparenting.msn.com



Kuva 4.1: NASA:n Pathfinder-projektin Marsin pinnalle lähettämä Sojouner-robotti, ja sen lähettämistä kuvista superresoluutiotekniikalla tuotettu kuva noin metrin kokoisesta kivenlohkareesta ("Yogi") Marsin pinnalla. Kuva löytyy osoitteesta http://mpfwww.jpl.nasa.gov/mpf/high-res.html.

tisesti tulkittavissa oleva malli, voidaan syntyviä malleja käyttää ongelmakentän ominaisuuksien analysoimisessa ja visualisoimisessa. NASAn Ames-tutkimuskeskuksen AutoClass-projektissa[6] on sovellettu bayesiläistä mallintamista menestyksellisesti mm. IRAS tähtikartaston kuvien, Yhdysvaltojen eri lentokenttien, ja eri DNA-sekvenssien analysoimisessa [24].

Kuvankäsittely NASAn tutkimusryhmät ovat kehittäneet menetelmän ns. *superresoluutiokuvien (super-resolution images)* tuottamiseksi. Menetelmässä on kyse useiden samasta kohteesta otettujen huonolaatuisten kuvien yhdistämisestä yhdeksi hyvälaatuiseksi ("superresoluutio"-) kuvaksi. Kuvassa 4.1 on tällainen Sojourner-robotin tuottamista kuvista bayesiläisellä tekniikalla muodostettu superresoluutiokuva kivenlohkareesta Marsin pinnalla.

Superresoluutiokuvia tuottava tekniikka ei perustu suoraan Bayes-verkkoteoriaan, mutta se seuraa puhtaasti edellä esitettyjä bayesiläisen mallintamisen periatteita (katso [23]). Paitsi Pathfinder-projektin aineistosta, NASAn tutkijat ovat tuottaneet superresoluutiokuvia myös Galileo-luotaimen aineistosta, sekä Viking-luotaimen Marsin pinnasta ottamista kuvista[7].

[6] URL: http:/ic-www.arc.nasa.gov/ic/projects/bayes-group/autoclass/
[7] URL: http://ic.arc.nasa.gov/ic/projects/super-res/

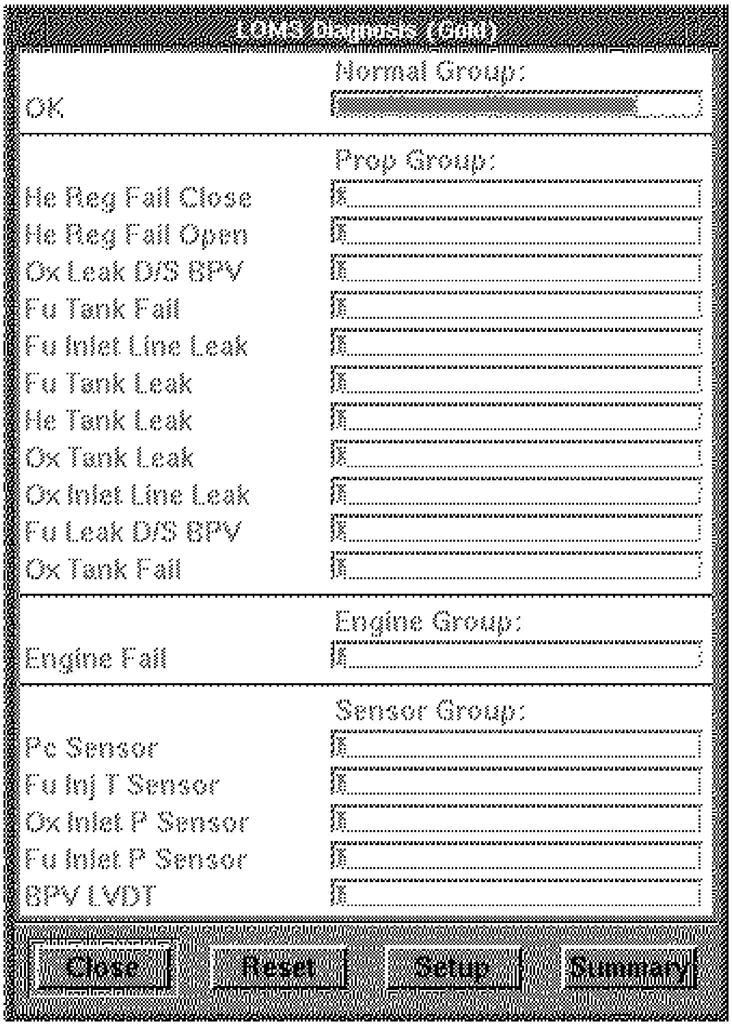
Hahmontunnistus Bayesiläistä mallinnusta on sovellettu hahmontunnistuksessa menestyksellisesti mm. NASA:n tutkimuskeskuksissa LandSat-satelliittien ottamien kuvien alueiden automaattisessa tunnistamisessa [64]. Käytetty Bayes-verkkomalli oli tässä tapauksessa luvussa 3.4 esitetty äärellinen sekajakaumamalli. Tavanomaisten Bayes-verkkojen käyttöä kuvantunnistuksessa käsitellään mm. lähteissä [61, 42]. Bayes-verkoilla on myös läheinen yhteys ns. *piilo-Markov-malleihin (hidden Markov models)* (katso [126, 30, 108]), joita käytetään yleisesti esimerkiksi puheentunnistukseen liittyvien hahmontunnistusongelmien ratkaisemisessa.

Vikadiagnostiikka Microsoftin tutkimusosasto on kehittänyt useita kymmeniä Bayes-verkkoihin perustuvia vikadiagnostiikkajärjestelmiä (troubleshooting wizards), jotka auttavat tietokoneen käyttäjää erilaisissa ongelmatilanteissa[8]. Intel käyttää Bayes-verkkoja prosessoripiirien valmistusprosessin laaduntarkkailussa ja American Airlines ohjelmistovirheiden etsimisessä[9].

Prosessikontrolli Yksi esimerkki prosessikontrolliin liittyvistä sovelluksista on kuvassa 3.4 esitetty ALARM-verkko jota käytetään sairaalapotilaiden tilaa tarkastelevan mittausprosessin ohjaamisessa [8]. NASA on kehittänyt laajan Bayes-verkkoihin perustuvan järjestelmän avaruussukkuloiden lentojen kontrollomiseksi. Järjestelmä paitsi ilmoittaa mahdollisista vikatilanteista, myös ehdottaa päätösteorian mukaisesti parhaat mahdolliset toimenpiteet kriisitilanteissa. Lisäksi ohjelmisto muokkaa aktiivisesti lentojen valvonnassa käytettäviä graafisia näyttöjä siten, että kullakin hetkellä näytetään vain tärkeimmät sukkuloiden tilaa kuvaavat mittaustulokset lentojen kulkua seuraavalle henkilökunnalle, mikä helpottaa teknikkojen työtä, ja nopeuttaa päätöksentekoa ongelmatilanteissa (katso kuva 4.2). VISTA-järjestelmä on ollut käytössä NASA:n Houstonin avaruussukkuloiden lennonjohtokeskuksessa jo useita vuosia. Enemmän tietoa VISTA-projektista löytyy projektin WWW-sivulta[10].

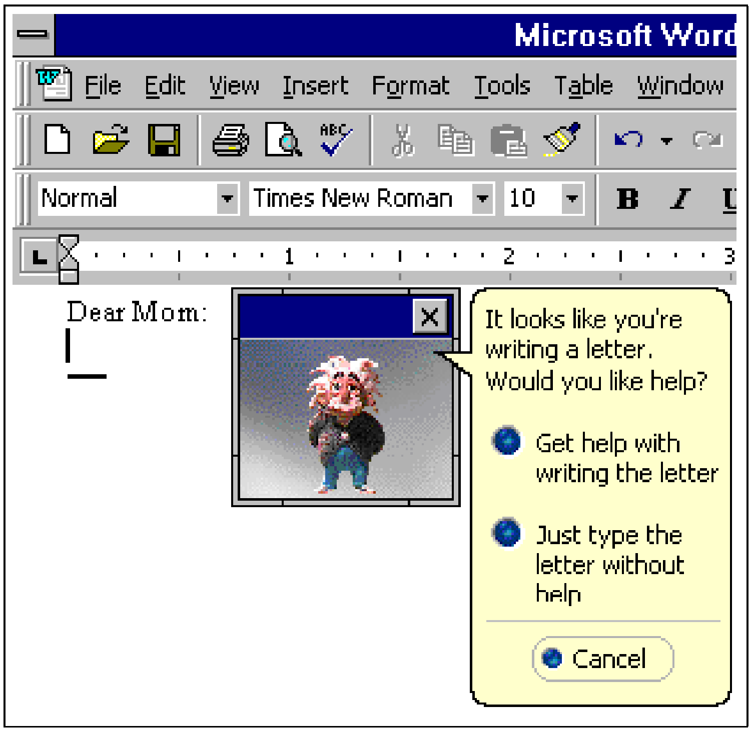
Bayes-verkkoja on sovellettu menestyksellisesti myös mm. lentokoneiden suihkumoottorien suunnittelussa (General Electric[11]), sekä erilaisten asejärjestelmien (Mitre/US Navy[12]) ja miehittämättömien vedenalaisten ajoneuvojen (Lockheed/DARPA[13]) ohjausprosesseissa.

[8] URL: http://support.microsoft.com/support/
[9] URL: http://www.auai.org/BN-Routine.html
[10] URL: http://www.research.microsoft.com/research/dtg/horvitz/vista.htm
[11] URL: http://www3.us.com/kic/kic/
[12] URL: http://www.afit.af.mil/Schools/EN/ENG/LABS/AI/BayesianNetworks/fieldedSystems.html#imsi
[13] URL: http://hugin.dk/lockheed-9604.html



Kuva 4.2: Esimerkki VISTA-järjestelmän muokkaamasta adaptiivisesta graafisesta näytöstä, jonka perusteella avaruussukkuloiden valvontakeskus seuraa lentojen kulkua.

Älykkäät agentit Älykkäät agentit ovat perussovellusohjelmiin liitettäviä itsenäisiä moduleita, jotka seuraavat järjestelmän toimintaa, ja helpottavat sovellusten käyttöä tarjoamalla aktiivisesti apua silloin kun katsovat sen käyttäjän kannalta tarpelliseksi. Microsoft on soveltanut Bayes-verkkotekniikkaa älykkäiden agenttien toteuttamisessa, ja tekniikka on jo käytössä Office'97-ohjelmistossa nimellä *Office Assistant*-nimisenä tuotteena (katso kuva 4.3). Microsoftin pitkän tähtäyksen suunnitelmana on toteuttaa Bayes-verkkotekniikoihin perustuva *adaptiivinen käyttöjärjestelmä*, koodinimeltään *Lumiere*, joka seu-



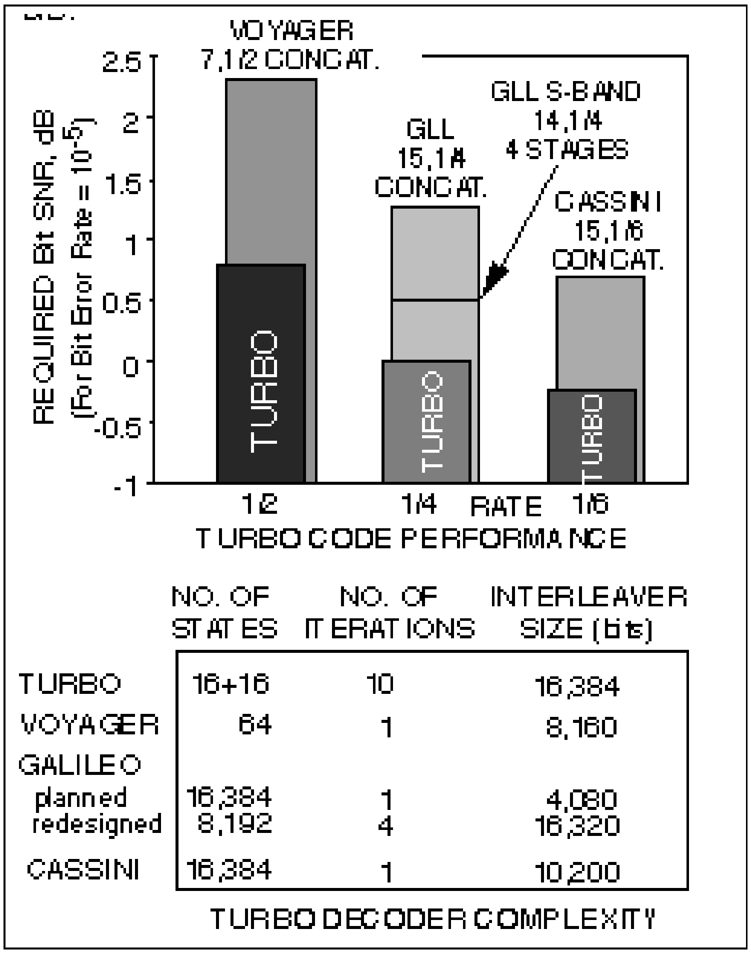
Kuva 4.3: Microsoftin Office Assistant-agentti.

raa tietokoneen käyttäjän toimintaa, ja adaptoituu toimimaan siten että tietokoneen käyttö on mahdollisimman tehokasta. Lisätietoja Lumiere-projektista löytyy projektin WWW-sivulta[14].

Tiedon kompressointi Kuten luvussa 2.4 näimme, bayesiläiselle lähestymistavalle voidaan antaa myös informaatioteoreettinen tulkinta todennäköisyysjakaumien ja koodinpituuksien läheisen yhteyden ansiosta. Niinpä ei olekaan yllättävää että Bayes-verkkoteoriaa voidaan soveltaa myös erilaisten koodausmenetelmien kehittämisessä, ja kehitettyjä koodeja voidaan soveltaa esimerkiksi tiedon kompressoinnissa. Vuonna 1993 esiteltiin uusi vallankumouksellinen koodausmenetelmä, *turbo-koodaus* [11], jonka on empiirisesti havaittu olevan huomattavasti tehokkaampi koodausmenetelmä kuin aikaisemmin käytetyt koodit (katso kuva 4.4[15]).

Teoksessa [90] osoitetaan kuinka turbo-koodaaja voidaan esittää Bayes-verkkomallina. Turbo-koodausta pidetään yhtenä tärkeimmistä koodausteo-

[14] URL: http://www.research.microsoft.com/research/dtg/horvitz/lum.htm
[15] Kuva löytyy WWW-sivulta URL: http://www331.jpl.nasa.gov/public/JPLtcodes.html



Kuva 4.4: Turbo-koodin vertailu avaruusluotaimissa nykyisin käytössä oleviin koodausmenetelmiin. Huomaa että histogrammikuvan asteikko on logaritminen, joten erot eri menetelmien välillä ovat huomattavia.

rian tuottamista tuloksista käytännön sovellusten kannalta. Menetelmä on erityisen kiinnostava avaruustutkimuksen kannalta, koska avaruusluotaimien lähetyskapasiteetti on melko pieni, jolloin tarvitaan tehokkaita tiedon pakkausalgoritmeja.

4.2 Bayes-verkkokehittimet

Koska uusia Bayes-verkko-ohjelmistoja syntyy kiihtyvässä tahdissa, oheinen lista saatavilla olevista Bayes-verkkokehittimistä vanhenee nopeasti. Elektronisesti saatavilla oleva Russell Almondin ylläpitämä lista löytyy Almondin kotisivulta[16], ja toinen vastaavanlainen lista CoSCo-ryhmän kotisivulta[17].

Ohjelmistot on seuraavassa jaoteltu kaupallisiin tuotteisiin ja tutkimusprototyyppeihin, mutta on huomattava että joistakin kaupallisista ohjelmistoista on saatavilla ilmainen versio tutkimus- tai opetuskäyttöön. Lisäksi useimmat kaupalliset yritykset tarjoavat tuotteistaan vapaasti kokeiltavia demo-versioita, jotka on saatavilla alla annettujen WWW-linkkien kautta. On myös todettava, että jotkut ilmaiseksi jaettavat, ja siksi tutkimusprototyypeiksi luokitellut ohjelmistot ovat ohjelmistoteknisesti kaupallisten ohjelmistojen tasolla (esim. Microsoftin MSBN).

Alla olevat ohjelmistot on jaoteltu sen mukaan, voidaanko ohjelmistoja käyttää Bayes-verkkojen oppimiseen datasta, vai tukevatko ohjelmat ainoastaan verkkojen manuaalista konstruomista. Oppimista tukevien ohjelmien lukumäärä ei ole toistaiseksi kovin suuri, mutta tilanne tullee muuttumaan lähitulevaisuudessa radikaalisti kun tuoreet aiheeseen liittyvät tutkimustulokset saadaan toteutettua sovellusasteelle.

4.2.1 Kaupalliset ohjelmistot

Oppimista tukevat ohjelmistot

- Ergo/Cogito
  Lisätietoja: Noetic Systems,
  URL: http://www.noeticsystems.com/ergo.shtml
- MIM
  Lisätietoja: David Edvards,
  URL: http://www.math.auc.dk/~jhb/CoCo/mim.html
- TETRAD
  Lisätietoja: Carnegie-Mellon University,
  URL: http://hss.cmu.edu/HTML/departments/philosophy/TETRAD/ tetrad.html

Manuaaliseen mallin konstruoimiseen perustuvat ohjelmistot

- Baron
  Lisätietoja: KC Associates,
  URL: mailto: kchang@gmu.edu
- Analytica
  Lisätietoja: Lumina Decision Systems,
  URL: http://www.lumina.com/software/index.html

[16]URL: http://bayes.stat.washington.edu/almond/belief.html
[17]URL: http://www.cs.Helsinki.FI/research/cosco/

- Decision Tools Suite
  Lisätietoja: Palisade,
  URL: http://www.palisade.com/

- DPL: Decision Programming Language
  Lisätietoja: Applied Decision Analysis,
  URL: http://www.dpl.adainc.com/

- DX Solution Series
  Lisätietoja: Knowledge Industries,
  URL: http://www.kic.com/

- GRAPHICAL-BELIEF
  Lisätietoja: Russell Almond,
  URL: http://bayes.stat.washington.edu/almond/gb/graphical-belief.html

- HUGIN
  Lisätietoja: Hugin Expert,
  URL: http://www.hugin.dk/

- NETICA
  Lisätietoja: Norsys Software,
  URL: http://www.norsys.com/netica.html

- STRATEGIST
  Lisätietoja: Prevision,
  URL: http://www.prevision.com/strategist.html

- DATA
  Lisätietoja: TreeAge Software,
  URL: http://www.treeage.com/

4.2.2 Tutkimusprototyypit

Oppimista tukevat ohjelmistot

- Belief Network Power Constructor
  Lisätietoja: Jie Cheng,
  URL: http://193.61.148.131/jcheng/bnpc.htm

- BIFROST
  Lisätietoja: Bo Thiesson,
  URL: http://www.sp.dk/~sorenh/software.html

- BKD: Bayesian Knowledge Discoverer
  Lisätietoja: Marco Ramoni,
  URL: http://kmi.open.ac.uk/~marco/projects/bkd/software/

- BNG
  Lisätietoja: Peter Haddawy,
  URL: http://www.cs.uwm.edu/public/dsail/research/bng.html

- BUGS
  Lisätietoja: Bugs-projekti,
  URL: http://www.mrc-bsu.cam.ac.uk/bugs/

- COCO
  Lisätietoja: Jens Henrik Badsberg,
  URL: http://www.math.auc.dk/~jhb/CoCo/cocoinfo.html

Manuaaliseen mallin konstruoimiseen perustuvat ohjelmistot

- BAYES
  Lisätietoja: Carnegie-Mellon University,
  URL: http://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/reasonng/probabl/bayes/0.html

- BAYONNET
  Lisätietoja: Yoichi Motomura,
  URL: http://www.etl.go.jp/etl/suri/motomura/BN/bn-java.html

- JAVABAYES
  Lisätietoja: Fabio Gozman,
  URL: http://www.cs.cmu.edu/~javabayes/Home/

- Bayesian Network Editor
  Lisätietoja: Sreekanth Nagarajan,
  URL: http://eel.engr.orst.edu/~nagarasr/bayesnet/index.html

- BELIEF
  Lisätietoja: Carnegie-Mellon University,
  URL: http://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/reasonng/probabl/belief/0.html

- IDEAL
  Lisätietoja: Rockwell International Science Center,
  URL: http://rpal.rockwell.com/ideal.html
  URL: mailto: ideal-request@rpal.rockwell.com

- MacEvidence
  Lisätietoja: Prakash Shenoy,
  URL: http://stat1.cc.ukans.edu:80/~pshenoy/

- Microsoft Bayes Networks (MSBN)
  Lisätietoja: Microsoft Research,
  URL: http://www.research.microsoft.com/research/dtg/msbn/

- Pulcinella
  Lisätietoja: IRIDIA,
  URL: http://iridia.ulb.ac.be/pulcinella/Welcome.html

- S-ElimBel
  Lisätietoja: Nicholas Thiéry,
  URL: http://www.spaces.uci.edu/thiery/elimbel/

- SPI
  Lisätietoja: Bruce D'Ambrosio,
  URL: http://www.cs.orst.edu/~dambrosi/

- TresBel
  Lisätietoja: Hong Xu,
  URL: http://iridia.ulb.ac.be/Projects/imple.html

- XBaies
  Lisätietoja: Robert Cowell,
  URL: mailto:xbaies-list@stats.ucl.ac.uk

- X-pert
  Lisätietoja: AI Group, Cantabria University,
  URL: http://ccaix3.unican.es/~AIGroup/

- XPress
  Lisätietoja: Mark Stitson,
  URL: mailto:M.Stitson@rhbnc.ac.uk

Yllä olevaan listaan on otettu mukaan vain standardeja Bayes-verkkoja käyttävät sovellukset. Latentteja muuttujia käyttäviä sekajakaumamalleja käyttäviä ohjelmistoja on lukuisia — tunnetuimpia näistä ohjelmistoista ovat:

- AutoClass
  Lisätietoja: NASAn AutoClass-projekti,
  URL: http://ic-www.arc.nasa.gov:80/ic/projects/bayes-group/group/autoclass/

- Snob
  Lisätietoja: Snob-kotisivu,
  URL: http://www.cs.monash.edu.au/~dld/Snob.html

Sekä AutoClass että Snob ovat tutkimusprototyyppejä, joiden toiminta perustuu mallien bayesiläiseen oppimiseen datasta. Linkkeja sekajakaumamallien kaupallisiin sovelluksiin, ja muihin tutkimusprototyyppeihin löytyy David Dowen ylläpitämältä WWW-sivulta[18].

Tilastollisista sovelluksistaan tunnettu SPSS-yhtiö on myös ryhtynyt soveltamaan bayesiläisiä tekniikoita kaupallisissa ohjelmistoissaan. Käytettävä malliperhe ei tosin ole Bayes-verkot, vaan neuroverkkomallit. Yhtiön mainoksen mukaan bayesiläisen lähestymistavan avulla voidaan välttää ylioppiminen, ja mallien oppiminen on myös myös yleensä nopeampaa. Lisätietoja bayesiläisiä neuroverkkoja soveltavasta "Neural Connection 2"-tuotteesta löytyy WWW-sivulta URL: http://www.spss.com/software/Neuro/. Vastaavanlainen vapaasti saatavilla olevilla ohjelmisto on raportoitu Radford Nealin väitöskirjassa [96], ja on saatavilla Nealin kotisivulta[19]. Nealin ohjelmistoa on laajennettu hiljakkoin käsittämään neuroverkkojen lisäksi myös äärellisiä sekajakaumamalleja.

[18] URL: http://www.cs.monash.edu.au/~dld/mixture.modelling.page.html
[19] URL: http://www.cs.utoronto.ca/~radford/

Luku 5

Tietolähteitä

Bayes-verkkojen aktiiviset kaupalliset soveltajat ja niitä markkinoivat yritykset. Bayes-verkkoja koskevaa kirjallisuutta. Tutkimusalan tutkimusryhmät ja järjestöt.

5.1 Kaupallinen sektori

Bayes-verkkojen kaupallistamisen pioneereja on tanskalainen HUGIN-yhtiö, joka soveltaa Aalborgin yliopistossa toimivan, professori Steffen Lauritzenin johtaman tutkimusryhmän tuloksia kaupallisen Hugin-ohjelmiston kehittämisessä. Kuten Bill Gatesin antamasta lausunnosta [54] saattaa arvata, merkittävin kaupallinen tekijä Bayes-verkkojen sovellusalueella on tällä hetkellä eittämättä Microsoft, jonka Bayes-verkkotutkimusryhmä on tutkimusalan suurimpia.

HUGIN-yhtiö on keskittynyt kehittämään ja myymään yhtä tuotetta, Hugin-kehitintä, kun taas Microsoft tuottaa sisäisiä sovelluksia, jotka liitetään yhtiön kaupallisiin tuotteisiin. Monet Bayes-verkkosovelluksia tekevistä yhtiöistä toimivat HUGIN-yhtiön tapaan tuottaen geneerisiä Bayes-verkkokehittimiä, mutta yhä useammat yhtiöt laajentavat toimintaansa myös konsultointialueelle, ja auttavat asiakkaitaan rakentamaan laajoihin järjestelmiin integroitavia räätälöityjä sovelluksia, samalla tavoin kuin Microsoft liittää oman tutkimuskeskuksensa tulokset tuotteisiinsa. Esimerkkejä tällaisista räätälöidyistä sovelluksista löytyy AUAI-järjestön kotivulta[1], samoin kuin lista syistä jotka ovat johtaneet Bayes-verkkojen käyttöönottoon[2].

Bayes-verkkotekniikoiden aktiivisiin kaupallisiin soveltajiin kuuluvat tällä hetkellä seuraavat yritykset:

[1] URL: http://www.auai.org/BN-Routine.html
[2] URL: http://www.auai.org/BN-Testimonial.html

- Applied Decision Analysis, URL: http://www.dpl.adainc.com/
- Decisioneering, URL: http://www.decisioneering.com/
- Hugin Expert, URL: http://www.hugin.dk/
- Knowledge Industries, URL: http://www.kic.com/
- Lumina Decision Systems, URL: http://www.lumina.com/
- Microsoft, URL: http://www.research.microsoft.com/research/dtg/
- Noetic Systems, URL: http://www.noeticsystems.com/
- Norsys Software, URL: http://www.norsys.com/home.html
- Palisade, URL: http://www.palisade.com/
- Prevision, URL: http://www.prevision.com/
- ThinkBank, URL: http://www.Thinkbank.COM/
- TreeAge Software, URL: http://www.treeage.com/
- Ultimode Systems, URL: http://www.ultimode.COM/index.html

5.2 Bayes-verkkotutkimus

5.2.1 Tutkimusryhmiä

Yksi merkittävimmistä Bayes-verkkotutkimusryhmistä on perinteisesti ollut professori Steffen Lauritzenin johtama tutkimusryhmä Aalborgin yliopistossa Tanskassa. Ryhmä on osallistunut aktiivisesti myös HUGIN-ohjelmistotuotteen (ks. luku 4.2) rakentamiseen. Muita merkittäviä tutkimusryhmiä Euroopassa löytyy mm. Suomesta, Belgiasta, ja Espanjasta (ks. allaoleva lista).

Maailmanlaajuisesti suurin tutkimusryhmä tällä hetkellä on luultavasti Microsoftilla, joka on haalinut koko 90-luvun ajan lupaavia tutkijoita tutkimuskeskukseensa. Microsoftin asema lienee vain vahvistumassa tulevaisuudessa, koska tutkimusryhmä on tuottanut jo useita menestyksellisiä Bayes-verkkosovelluksia (ks. luku 4.1), ja yhtiö on lisäksi lisäämässä tutkimusrahoitustaan moninkertaiseksi nykyiseen verrattuna. Yrityksen pääjohtajan Bill Gatesin lausunnot [54] kuvaavat myös vahvaa luottamusta Bayes-verkkotekniikkaan. Toinen merkittävä tekijä tutkimusalueella Yhdysvalloissa on NASA, jolla on ollut lukuisia menestyksellisiä Bayes-verkkoprojekteja.

Monash-yliopiston tutkimusryhmä Australiassa on tehnyt pioneerityötä sekajakaumamallien tutkimuksen alueella, samoin kuin CoSCo-ryhmä Helsingin yliopistossa.

Alla oleva lista sisältää joitakin WWW-linkkejä Bayes-verkkotutkimusta tekeviin ryhmiin Suomessa, Euroopassa, ja muualla maailmassa. Lista sisältää luonnollisesti vain muutaman esimerkin Bayes-verkkotutkimusta tekevistä tutkimusryhmistä; lisää hyödyllisiä WWW-linkkejä (mm. yli 100 nimeä käsittävä luettelo bayesiläistä mallintamista tutkivista henkilöistä[3]) löytyy AUAI-järjestön kotisivulta (ks. luku 5.2.4).

- Helsingin yliopisto
  URL: http://www.cs.Helsinki.FI/research/cosco/
- Aalborgin yliopisto,
  URL: http://www.cs.auc.dk/general/ES/
- Cantabrian yliopisto/Cornell University
  URL: http://ccaix3.unican.es/~AIGroup/
- IRIDIA,
  URL: http://iridia.ulb.ac.be/Projects/imple.html
- NASA
  URL: http://fi-www.arc.nasa.gov/fia/projects/bayes-group/
- Microsoft
  URL: http://www.research.microsoft.com/research/dtg/
- Monash University
  URL: http://www.cs.monash.edu.au/~dld/

5.2.2 Kirjallisuutta

Bayes-verkkokirjallisuuden klassikkona pidetään Pearlin vuonna 1988 julkaisemaa kirjaa [105], joka vielä hieman vanhentuneenakin sopii hyväksi johdatukseksi Bayes-verkkoteoriaan. Pearlin lähestymistapa perustuu pitkälti verkkoteoreettisiin käsitteisiin, kun taas lähes samaan aikaan julkaistu Neapolitanin oppikirja [97] tarjoaa matemaattisesti hieman pragmaattisemman, asiantuntijajärjestelmien rakentamiseen tähtäävän yleisesityksen. Charniakin artikkeli [19] antaa selkeän, ei-matemaattisen johdatuksen Bayes-verkkojen peruskäsitteisiin. Teknisempi lyhyt katsaus aiheeseen löytyy artikkelista [129].

[3] URL: http://bayes.stat.washington.edu/bayes_people.html

Edellä mainitut artikkelit, samoin kuin Pearlin ja Neapolitanin kirjat, ovat hieman vanhentuneita mm. siinä mielessä, että niissä ei juurikaan käsitellä Bayes-verkkojen oppimista. Uudempia johdatuksia Bayes-verkkoteoriaan tarjoavat Spirtesin et al. [130], Jensenin [60], Schaferin [118], ja Castillon et al. [18] kirjat, sekä Heckermanin artikkeli [49]. Spirtesin kirja käsittelee kausaalisia Bayes-verkkoja, ja niiden ei-bayesiläistä oppimista. Jensenin ja Schaferin kirjat keskittyvät kuvaamaan probabilistisen päättelyn toteuttamista Bayes-verkoissa, eivätkä käsittele juuri lainkaan Bayes-verkkojen oppimista. Castillon kirja on selkeää luettavaa, mutta Bayes-verkkojen oppimista käsittelevä luku sisältää joitakin epätarkkuuksia. Heckermanin artikkeli on hyvä yleiskatsaus Bayes-verkkotutkimuksen tuoreimpiin tuloksiin. Lauritzenin [84] ja Whittakerin [141] kirjat käsittelevät yleisemmin graafisiin verkkoesityksiin perustuvan mallintamisen teoriaa.

Bayes-verkkojen oppimiseen keskittyvää materiaalia löytyy lähteistä [50, 48, 16, 17]. Suomenkielisiä katsauksia tutkimusalueeseen löytyy tutkielmista [122, 5]. Yleisemmin bayesiläistä mallinmuodostusta käsitellään lähteissä [22, 88, 96]. Näistä Mackay [88] ja Neal [96] keskittyvät bayesiläisen lähestymistavan soveltamiseen neuroverkkojen oppimisessa. Muita neuroverkkojen ja bayesiläisen mallintamisen välisiä yhteyksiä käsitteleviä teoksia ovat [13, 14, 93, 117].

Teoreettisempia katsauksia todennäköisyyslaskennan soveltamistekniikoihin löytyy teoksista [41, 10, 59]. Hyviä johdatuksia päätösteoriaan löytyy lähteistä [32, 9, 38, 86]. Epätäsmällisyyden käsitettä yleisemmin käsitellään teoksissa [119, 102, 138]. Todennäköisyyslaskennan ja sumean logiikan suhdetta tarkastellaan artikkeleissa [21, 20].

Lisää viitteitä Bayes-verkkoihin liittyviin teoksiin löytyy seuraavilta WWW-sivuilta:

- URL: http://www.cs.pitt.edu/~tsamard/bnpointers.html
- URL: http://www.cs.cmu.edu/afs/cs.cmu.edu/user/ldc/WWW/bayes-net-research/bayes-net-research.html
- URL: http://www.Ultimode.com/~wray/graphbib/
- URL: http://www.lis.pitt.edu/~dsl/da-books.html

Elektronisesti luettavia (HTML-formaatissa olevia) johdatuksia Bayes-verkkoteoriaan löytyy mm. seuraavilta WWW-sivuilta:

- URL: http://www.afit.af.mil/Schools/EN/ENG/LABS/AI/BayesianNetworks/

- URL: http://www.cm.cf.ac.uk/Dave/AI2/AI_notes.html
- URL: http://www.research.att.com/~dmac/notes/bayesian.html
- URL: http://yoda.cis.temple.edu:8080/UGAIWWW/lectures97/uncertainty/bindex.html

Lisäksi useiden Bayes-verkkotutoriaalien/kurssien kalvot ovat myös saatavilla elektronisessa muodossa. Näitä löytyy mm. seuraavilta WWW-sivuilta:

- URL: http://www.cs.Helsinki.FI/~myllymak
- URL: http://www.cs.Helsinki.FI/~tirri
- URL: http://www.research.microsoft.com/research/dtg/heckerma/heckerma.html
- URL: http://www.ultimode.COM/~wray/
- URL: http://www.auai.org/auai-tutes.html

5.2.3 Konferensseja ja lehtiä

Yksinomaan Bayes-verkkotutkimukseen keskittynyttä jokavuotista tieteellistä konferenssia ei toistaiseksi ole perustettu, joskin *Uncertainty in Artificial Intelligence (UAI)*-konferenssit ovat käytännössä muodostuneet sellaiseksi — lähes kaikki konferenssissa nykyään julkaistavat artikkelit käsittelevät Bayes-verkkoja. Bayesiläiseen mallinnukseen yleisesti liittyviä asioita käsitteleviä konferensseja ovat mm. *International Workshops on Artificial Intelligence and Statistics*, *Valencia International Meetings on Bayesian Statistics*, *International FLAIRS Conferences (Special Track on Uncertain Reasoning)*, ja *International Workshops on Maximum Entropy and Bayesian Methods.* Koska bayesiläinen mallinnus liittyy läheisesti koneoppimiseen ja muihin tekoälyn osa-alueisiim, julkaistaan alueeseen liittyviä artikkeleita paljon mm. seuraavissa konferensseissa: *Knowledge Discovery and Data mining (KDD)*, *International Conferences on Machine Learning (ICML)*, *International Joint Conferences on Artificial Intelligence (IJCAI)*, ja *National Conferences on Artificial Intelligence (AAAI).* Neuroverkkokonferensseista ovat jokavuotiset *Neural Information Processing Systems (NIPS)*-konferenssit julkaisseet yhä kasvavassa määrin bayesiläiseen mallinnukseen liittyviä artikkeleita. Eurooppalaiset epätäsmällisen tiedon käsittelyyn liittyvät konferenssit, kuten esimerkiksi *International Joint Conferences on Qualitative and Quantitative Practical Reasoning (ECSQARU)*, *European Symposiums on Intelligent*

Techniques, ja *The European Congresses on Intelligent Techniques and Soft Computing (EUFIT)*, ovat keskittyneet pääasiassa ei-bayesiläisten lähestymistapojen käsittelemiseen.

Bayes-verkkotutkijat eivät toistaiseksi ole perustaneet omaa, Bayes-verkkotutkimukselle omistautunutta tieteellistä julkaisua, vaan alan tutkimustulokset ilmestyvät monissa eri tekoälyn osa-alueita käsittelevissä lehdissä, joita ovat mm. *Artificial Intelligence*, *Journal of Artificial Intelligence Research*, *Journal of Approximate Reasoning*, ja *Machine Learning*.

5.2.4 Järjestöjä

Koska edellä mainittu UAI-konferenssi on muodostunut Bayes-verkkotutkimusalueen tärkeimmäksi vuosittaiseksi tapaamiseksi, on konferenssin järjestävästä *Association for Uncertainty in Artificial Intelligence (AUAI)*-organisaatiosta luonnollisesti muovautunut eräänlainen Bayes-verkkotutkijoiden kattojärjestö. Muita merkittäviä alan tutkimukseen liittyviä järjestöjä ovat *International Society for Bayesian Analysis (ISBA)*, joka sponsoroi *Valencia International Meeting on Bayesian Statistics*- ja *International Workshop on Maximum Entropy and Bayesian Methods*-kokouksia, sekä *Society for Artificial Intelligence and Statistics*, joka järjestää joka toinen vuosi pidettävän *International Workshop on Artificial Intelligence and Statistics*-kokouksen. Muita tutkimusalueeseen liittyviä järjestöjä on lueteltu alla. Eurooppalaisista organisaatioista on huomautettava, että kuten eurooppalaiset alan konferenssit, myös tieteelliset organisaatiot (kuten esim. ERUDIT, NeuroCOLT, ja MLnet) ovat keskittyneet pääasiassa ei-bayesiläisiin epätäsmällisyyden käsittelytapoihin.

- Association for Uncertainty in Artificial Intelligence (AUAI),
  URL: http://www.auai.org/
- Society for Artificial Intelligence and Statistics,
  URL: http://www.vuse.vanderbilt.edu/~dfisher/ai-stats/society.html
- International Society for Bayesian Analysis (ISBA)
  URL: http://omega.albany.edu:8008/isba/
- ASA Section on Bayesian Statistical Sciences
  URL: http://www.isds.duke.edu/sbss/sbss.html
- European Association for Decision Making
  URL: http://huizen.dds.nl/~eadm/

- Decision Analysis Society
  URL: http://www.fuqua.duke.edu/faculty/daweb/
- Society for Risk Analysis (SRA)
  URL: http://www.sra.org/
- Classification Society of North America (CSNA)
  URL: http://www.pitt.edu/~csna/
- Network of Excellence for Uncertainty Modeling and Fuzzy Technology in the European Union (ERUDIT)
  URL: http://ss-m3.mitgmbh.de/erudit/
- ESPRIT Working Group in Neural and Computational Learning (NeuroCOLT)
  URL: http://www.cs.rhbnc.ac.uk/research/compint/neurocolt/
- ESPRIT Network of Excellence in Machine Learning (MLnet)
  URL: http://omega.gmd.de/ml-archive/MLnet/MLnet.html

Kirjallisuutta

[1] Aha, D., Ed. *Lazy Learning*. Kluwer Academic Publishers, Dordrecht, 1997. Reprinted from Artificial Intelligence Review, 11:1–5.

[2] Aha, D., Kibler, D., and Albert, M. Instance-based learning algorithms. *Machine Learning 6* (1991), 37–66.

[3] Alag, S., and Agogino, A. Inference using message propagation and topology transformation in vector gaussian continuous networks. In *Proceedings of the Twelfth Annual Conference on Uncertainty in Artificial Intelligence (UAI–96)* (Portland, Oregon, 1996), pp. 20–27.

[4] Amari, S., Murata, N., Müller, K.-R., Finker, M., and Hua Yang, H. Asymptotic statistical theory of overtraining and cross-validation. *IEEE Transactions on Neural Networks 8*, 5 (February 1997), 985–996.

[5] Antti-Poika, T. Bayes-verkkojen käyttö luokittelussa. Master's thesis, Deparment of Computer Science, University of Helsinki, 1997.

[6] Baxter, R., and Oliver, J. MDL and MML: Similarities and differences. Tech. Rep. 207, Department of Computer Science, Monash University, 1994.

[7] Becker, A., and Geiger, D. A sufficiently fast algorithm for finding close to optimal junction trees. In *Proceedings of the Twelfth Annual Conference on Uncertainty in Artificial Intelligence (UAI–96)* (Portland, Oregon, 1996), pp. 81–89.

[8] Beinlich, I., Suermondt, H., Chavez, R., and Cooper, G. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *Proceedings of the Second European Conf. on Artificial Intelligence in Medicine* (London, UK, August 1989), pp. 247–256.

[9] Berger, J. *Statistical Decision Theory and Bayesian Analysis.* Springer-Verlag, New York, 1985.

[10] Bernardo, J., and Smith, A. *Bayesian theory.* John Wiley, 1994.

[11] Berrou, G., Glavieux, A., and Thitimajshima, P. Near Shannon limit error-correcting coding: Turbo codes. In *Proceedings of the 1993 International Conference in Communications* (geneve, May 1993), pp. 1064–1070.

[12] Bishop, C. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[13] Bishop, C., Svensén, M., and Williams, C. EM optimization of latent-variable density models. In *Advances in Neural Information Processing Systems 8*, D. Touretzky, M.C.Mozer, and M.E.Hasselmo, Eds. MIT Press, 1996.

[14] Bishop, C., Svensén, M., and Williams, C. GTM: A principled alternative to the self-organizing map. In *Advances in Neural Information Processing Systems 9 (to appear).* MIT Press, 1997.

[15] Breiman, L. Bias, variance, and arcing classifiers. Tech. Rep. TR 460, University of California, Statistics Department, April 1996.

[16] Buntine, W. Operations for learning with graphical models. *Journal of Artificial Intelligence Research 2* (1994), 159–225.

[17] Buntine, W. A guide to the literature on learning graphical models. *IEEE Transactions on Knowledge and Data Engineering 8* (1996), 195–210.

[18] Castillo, E., Gutiérrez, J., and Hadi, A. *Expert Systems and Probabilistic Network Models.* Monographs in Computer Science. Springer-Verlag, New York, NY, 1997.

[19] Charniak, E. Bayesian networks without tears. *AI Magazine 12*, 4 (1991), 50–63.

[20] Cheeseman, P. Probabilistic versus fuzzy reasoning. In Kanal and Lemmer [63], pp. 85–102.

[21] Cheeseman, P. In defense of probability. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence* (1995), vol. 2, pp. 1002–1009.

[22] Cheeseman, P. On Bayesian model selection. In *The Mathematics of Generalization*, D. Wolpert, Ed., vol. XX of *SFI Studies in the Sciences of Complexity*. Addison-Wesley, 1995, pp. 315–330.

[23] Cheeseman, P., Kanefsky, B., Hanson, R., and Stutz, J. Super-resolved surface reconstruction from multiple images. Tech. Rep. FIA-94-12, NASA Ames Research Center, Artificial Intelligence Branch, October 1994.

[24] Cheeseman, P., and Stutz, J. Bayesian classification (AutoClass): Theory and results. In *Advances in Knowledge Discovery and Data Mining*, U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, Eds. AAAI Press, Menlo Park, 1996, ch. 6.

[25] Chickering, D., Geiger, D., and Heckerman, D. Learning Bayesian networks is NP-hard. Tech. Rep. MSR-TR-94-17, Microsoft Research, 1994.

[26] Chickering, D., and Heckerman, D. Efficient approximations for the marginal likelihood of incomplete data given a Bayesian network. In *Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence* (Portland, Oregon, August 1996), E. Horvitz and F. Jensen, Eds., Morgan Kaufmann Publishers, pp. 158–168.

[27] Cooper, G. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence 42*, 2–3 (March 1990), 393–405.

[28] Cooper, G., and Herskovits, E. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning 9* (1992), 309–347.

[29] Dagum, P., and Luby, M. Approximating probabilistic inference in Bayesian belief networks is NP-hard. *Artificial Intelligence 60* (1993), 141–153.

[30] Dean, T., and Wellman, M. *Planning and Control*. Morgan Kaufmann Publishers, San Mateo, CA, 1991.

[31] Dechter, R., and Rish, I. A scheme for approximating probabilistic inference. In *Proceedings of the Thirteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI–97)* (San Francisco, CA, 1997), Morgan Kaufmann Publishers, pp. 132–141.

[32] DeGroot, M. *Optimal statistical decisions.* McGraw-Hill, 1970.

[33] Dempster, A., Laird, N., and Rubin, D. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B 39*, 1 (1977), 1–38.

[34] Druzdzel, M., and Henrion, M. Efficient reasoning in qualitative probabilistic networks. In *Proceedings of the Eleventh National Conference on Artificial Intelligence* (Washington, DC, July 1993), AAAI Press/MIT, Menlo Park, CA, pp. 548–553.

[35] Everitt, B., and Hand, D. *Finite Mixture Distributions.* Chapman and Hall, London, 1981.

[36] Friedman, J. On bias, variance, 0/1–loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery 1*, 1 (1997), 55–78.

[37] Friedman, N., and Goldszmidt, M. Discretizing continuous attributes while learning Bayesian networks. In *Machine Learning: Proceedings of the Thirteenth International Conference* (1996), L. Saitta, Ed., Morgan Kaufmann Publishers, pp. 157–165.

[38] Gärdenfors, P., and Sahlin, N.-E., Eds. *Decision, Probability, and Utility.* Cambridge University Press, New York, 1988.

[39] Geiger, D., and Heckerman, D. A characterization of the Dirichlet distribution through global and local independence. Tech. Rep. MSR-TR-94-16, Microsoft Research, November (revised February 1995) 1994.

[40] Geiger, D., and Heckerman, D. Learning bayesian networks. Tech. Rep. MSR-TR-95-02, Microsoft Research, December 1994.

[41] Gelman, A., Carlin, J., Stern, H., and Rubin, D. *Bayesian Data Analysis.* Chapman & Hall, 1995.

[42] Gong, S., and Buxton, H. Bayesian nets for mapping contextual knowledge to computational constraints in motion segmentation and tracking. In *British machine vision conference* (Guildford, Surrey, UK, September 1993).

[43] Gyllenberg, M., Koski, T., and Verlaan, M. Classification of binary vectors by stochastic complexity. Tech. Rep. A5, University of Turku, Institute for Applied Mathematics, November 1994.

[44] HASSOUN, M. *Fundamentals of Artificial Neural Networks.* MIT Press, Cambridge, Massachusetts, 1995.

[45] HECHT-NIELSEN, R. *Neurocomputing.* Addison-Wesley Publishing Company, Reading, MA, 1990.

[46] HECKERMAN, D. Probabilistic interpretation for MYCIN's certainty factors. In Kanal and Lemmer [63], pp. 167–196.

[47] HECKERMAN, D. A Bayesian approach to learning causal network. In *Uncertainty in Artificial Intelligence: Proceedings of the Eleventh Conference* (Montreal, Canada, 1995), P. Besnard and S. Hanks, Eds., pp. 285–295.

[48] HECKERMAN, D. A tutorial on learning with Bayesian networks. Tech. Rep. MSR-TR-95-06, Microsoft Research, Advanced Technology Division, One Microsoft Way, Redmond, WA 98052, 1996.

[49] HECKERMAN, D. Bayesian networks for data mining. *Data Mining and Knowledge Discovery 1*, 1 (1997), 79–119.

[50] HECKERMAN, D., GEIGER, D., AND CHICKERING, D. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning 20*, 3 (September 1995), 197–243.

[51] HECKERMAN, D., AND NATHWANI, B. An evaluation of the diagnostic accuracy of Pathfinder. *Computers and Biomedical Research 25* (1992), 56–74.

[52] HECKERMAN, D., AND SCHACHTER, R. Decision-theoretic foundations for causal reasoning. *Journal of Artificial Intelligence Research 3* (1995), 405–430.

[53] HECKERMAN, D., AND SHWE, M. Diagnosis of multiple faults: A sensitivity analysis. In *Uncertainty in Artificial Intelligence 9*, D. Heckerman and A. Mamdani, Eds. Morgan Kaufmann Publishers, San Mateo, CA, 1993, pp. 80–87.

[54] HELM, L. The future of software may lie in the obscure theories of an 18th century cleric called Thomas Bayes. *Los Angeles Times* (Monday, October 28 1996).

[55] HENRION, M. Uncertainty in artificial intelligence: Is probability epistemologically and heuristically adequate? In *Expert Judgements and*

Expert Systems, J. Mumpower, L. Philipps, O. Renn, and U. V., Eds., NATO ASI Series F: Computer and Systems Science. Springer-Verlar, Berlin, 1987, pp. 106–129.

[56] Henrion, M. An introduction to algorithms for inference in belief nets. In *Uncertainty in Artificial Intelligence 5*, M. Henrion, R. Shachter, L. Kanal, and J. Lemmer, Eds. Elsevier Science Publishers B.V. (North-Holland), Amsterdam, 1990, pp. 129–138.

[57] Henrion, M., Pradhan, M., Del Favero, B., Huang, K., Provan, G., and O'Rorke, P. Why is diagnosis using belief networks insensitive to imprecision in probabilities? In *Uncertainty in Artificial Intelligence, Proceedings of the Twelfth Conference* (Portland, Oregon, August 1996), E. Horvits and F. Jensen, Eds., Morgan Kaufmann Publishers, San Francisco, CA, pp. 307–314.

[58] Isomursu, P., Niskanen, V., Carlsson, C., and Eklund, P. Sumean logiikan mahdollisuudet. Tech. Rep. 34/93, Teknologian kehittämiskeskus (TEKES), 1993.

[59] Jaynes, E. Probability theory: The logic of science. Under construction. A draft copy can be dowloaded from 'ftp://bayes.wustl.edu/Jaynes.book'., 1996.

[60] Jensen, F. *An Introduction to Bayesian Networks.* UCL Press, London, 1996.

[61] Jensen, F., Christensen, H., and Nielsen, J. Bayesian methods for interpretation and control in multi-agent vision systems. *Applications of Artificial Intelligence X: Machine Vision and Robotics, SPIE Proceedings Series 1708* (1992).

[62] Jordan, M., and Jacobs, R. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation 6* (1994), 181–214.

[63] Kanal, L., and Lemmer, J., Eds. *Uncertainty in Artificial Intelligence 1.* Elsevier Science Publishers B.V. (North-Holland), Amsterdam, 1986.

[64] Kanefsky, B., Stutz, J., Cheeseman, P., and Taylor, W. An improved automatic classification of a Landsat/TM image from Kansas (FIFE). Tech. Rep. FIA-94-01, NASA Ames Research Center, Artificial Intelligence Branch, May 1994.

[65] Kass, R., and Raftery, A. Bayes factors. Tech. Rep. 254, Department of Statistics, University of Washington, 1994.

[66] Kjærulff, U. Optimal decomposition of probabilistic networks by simulated annealing. *Statistics and Computing 2* (1992), 7–17.

[67] Kohonen, T. *Self-Organizing Maps.* Springer-Verlag, Berlin, 1995.

[68] Kolodner, J. *Case-Based Reasoning.* Morgan Kaufmann Publishers, San Mateo, 1993.

[69] Kontkanen, P., Myllymäki, P., Silander, T., and Tirri, H. Bayes optimal lazy learning. Tech. Rep. NC-TR-97-031, ESPRIT Working Group on Neural and Computational Learning (NeuroCOLT), 1997.

[70] Kontkanen, P., Myllymäki, P., Silander, T., and Tirri, H. A Bayesian approach for retrieving relevant cases. In *Artificial Intelligence Applications (Proceedings of the EXPERSYS-97 Conference)* (Sunderland, UK, October 1997), P. Smith, Ed., IITT International, pp. 67–72.

[71] Kontkanen, P., Myllymäki, P., Silander, T., and Tirri, H. A Bayesian approach to discretization. In *Proceedings of the European Symposium on Intelligent Techniques* (Bari, Italy, March 1997), pp. 265–268.

[72] Kontkanen, P., Myllymäki, P., Silander, T., and Tirri, H. Comparing stochastic complexity minimization algorithms in estimating missing data. In *Proceedings of WUPES'97, the 4th Workshop on Uncertainty Processing* (Prague, Czech Republic, January 1997), pp. 81–90.

[73] Kontkanen, P., Myllymäki, P., Silander, T., and Tirri, H. On the accuracy of stochastic complexity approximations. In *Proceedings of the Causal Models and Statistical Learning Seminar* (London, UK, March 1997), pp. 103–117.

[74] Kontkanen, P., Myllymäki, P., Silander, T., Tirri, H., and Grünwald, P. Comparing predictive inference methods for discrete domains. In *Proceedings of the Sixth International Workshop on Artificial Intelligence and Statistics* (Ft. Lauderdale, Florida, January 1997), pp. 311–318.

[75] Kontkanen, P., Myllymäki, P., Silander, T., Tirri, H., and Grünwald, P. On predictive distributions and Bayesian networks. In *Proceedings of the Seventh Belgian-Dutch Conference on Machine Learning (BeNeLearn'97)* (Tilburg, the Netherlands, October 1997), W. Daelemans, P. Flach, and A. van den Bosch, Eds., pp. 59–68.

[76] Kontkanen, P., Myllymäki, P., Silander, T., Tirri, H., and Grünwald, P. On predictive distributions and Bayesian networks. Tech. Rep. NC-TR-97-032, ESPRIT Working Group on Neural and Computational Learning (NeuroCOLT), 1997.

[77] Kontkanen, P., Myllymäki, P., and Tirri, H. Comparing Bayesian model class selection criteria by discrete finite mixtures. In *Information, Statistics and Induction in Science* (Proceedings of the ISIS'96 Conference, Melbourne, Australia, August 1996), D. Dowe, K. Korb, and J. Oliver, Eds., World Scientific, Singapore, pp. 364–374.

[78] Kontkanen, P., Myllymäki, P., and Tirri, H. Constructing Bayesian finite mixture models by the EM algorithm. Tech. Rep. NC-TR-97-003, ESPRIT Working Group on Neural and Computational Learning (NeuroCOLT), 1996.

[79] Kontkanen, P., Myllymäki, P., and Tirri, H. Predictive data mining with finite mixtures. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (Portland, Oregon, August 1996), E. Simoudis, J. Han, and U. Fayyad, Eds., pp. 176–182.

[80] Kontkanen, P., Myllymäki, P., and Tirri, H. Some experimental results with finite mixture models. In *Book of Abstracts, First European Conference on Highly Structured Stochastic Systems* (Rebild, Denmark, May 1996), pp. 112–115.

[81] Kontkanen, P., Myllymäki, P., and Tirri, H. Experimenting with the Cheeseman-Stutz evidence approximation for predictive modeling and data mining. In *Proceedings of the Tenth International FLAIRS Conference* (Daytona Beach, Florida, May 1997), D. Dankel, Ed., pp. 204–211.

[82] Langley, P., Iba, W., and Thompson, K. An analysis of Bayesian classifiers. In *Proceedings of the 10th National Conference on Artificial Intelligence* (San Jose, CA, July 1992), MIT Press, pp. 223–228.

[83] Lauritzen, S. Propagation of probabilities, means and variances in mixed graphical association models. *Journal of the American Statistical Association 87* (1992), 1098–1108.

[84] Lauritzen, S. *Graphical Models.* Oxford University Press, 1996.

[85] Li, Z., and D'Ambrosio, B. Efficient inference in bayes nets as a combinatorial optimization problem. *International Journal of Approximate Reasoning 11*, 1 (1994), 55–81.

[86] Lindley, D. *Making Decisions*, second ed. John Wiley & Sons, London, 1992.

[87] M., W., and Holyoak, K. Predictive and diagnostic learning within causal models: Asymmetries of cue competition. *Journal of Experimental Psychology: General 121* (1992), 222–236.

[88] Mackay, D. *Bayesian Methods for Adaptive Models.* PhD thesis, California Institute of Technology, 1992.

[89] MacKay, D. J. C. Bayesian non-linear modelling for the prediction competition. In *ASHRAE Transactions, V.100, Pt.2* (Atlanta Georgia, 1994), ASHRAE, pp. 1053–1062.

[90] McEliece, R., Mackay, D., and Cheng, J.-F. Turbo decoding as an instance of Pearl's "belief propagation" algorithm. *International Journal of Selected areas in Communication* (1997). (to appear).

[91] McLachlan, G., and Thriyambakam, K., Eds. *The EM Algorithm and Extensions.* John Wiley & Sons, New York, 1997.

[92] Moody, J., and Darken, C. Fast learning in networks of locally-tuned processing units. *Neural Computation 1* (1989), 281–294.

[93] Myllymäki, P. *Mapping Bayesian Networks to Stochastic Neural Networks: A Foundation for Hybrid Bayesian-Neural Systems.* PhD thesis, Report A-1995-1, Department of Computer Science, University of Helsinki, December 1995.

[94] Myllymäki, P., and Tirri, H. Bayesian case-based reasoning with neural networks. In *Proceedings of the IEEE International Conference on Neural Networks* (San Francisco, March 1993), vol. 1, IEEE, Piscataway, NJ, pp. 422–427.

[95] Myllymäki, P., and Tirri, H. Massively parallel case-based reasoning with probabilistic similarity metrics. In *Topics in Case-Based Reasoning*, S. Wess, K.-D. Althoff, and M. Richter, Eds., vol. 837 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, 1994, pp. 144–154.

[96] Neal, R. *Bayesian Learning for Neural Networks*. No. 118 in Lecture Notes in Statistics. Springer-Verlag, New York, NY, 1996.

[97] Neapolitan, R. *Probabilistic Reasoning in Expert Systems*. John Wiley & Sons, New York, NY, 1990.

[98] Oliver, J., and Baxter, R. MML and Bayesianism: Similarities and differences. Tech. Rep. 206, Department of Computer Science, Monash University, December 1994.

[99] Oliver, J., and Hand, D. Introduction to minimum encoding inference. Tech. Rep. 205, Department of Computer Science, Monash University, July 1994.

[100] Oliver, J., and Hand, D. On pruning and averaging decision trees. In *Machine Learning: Proceedings of the Twelfth International Conference* (1995), A. Prieditis, Ed., Morgan Kaufmann Publishers, pp. 430–437.

[101] Oliver, R., and Smith, J. *Influence diagrams, belief nets and decision analysis*. John Wiley & Sons, 1990.

[102] Paris, J. *The Uncertain Reasoner's Companion: A Mathematical Prespective*, vol. 39 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, 1994.

[103] Parsa, I. KDD-97 knowledge discovery and data mining tools competition. *Knowledge Discovery Nuggets 97*, 19 (1997). An electronic journal, available at `http://www.kdnuggets.com/`.

[104] Pearl, J. Fusion, propagation and structuring in belief networks. *Artificial Intelligence 29* (1986), 241–288.

[105] Pearl, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, San Mateo, CA, 1988.

[106] Pearl, J., and Verma, T. A theory of inferred causation. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference* (1991), A. Allen, R. Fikes, and E. Sandewall, Eds., Morgan Kaufmann Publishers, San Mateo, CA, pp. 441–452.

[107] Poggio, T., and Girosi, F. Networks for approximation and learning. *Proceedings of the IEEE 78*, 9 (1990), 1481–1497.

[108] Poland, W. *Decision Analysis with Continuous and Discrete Variables: A Mixture Distribution Approach.* PhD thesis, Department of Engineering Economic Systems, Stanford University, 1994.

[109] Rissanen, J. Stochastic complexity. *Journal of the Royal Statistical Society 49*, 3 (1987), 223–239 and 252–265.

[110] Rissanen, J. *Stochastic Complexity in Statistical Inquiry.* World Scientific Publishing Company, New Jersey, 1989.

[111] Rissanen, J. Fisher information and stochastic complexity. *IEEE Transactions on Information Theory 42*, 1 (January 1996), 40–47.

[112] Rissanen, J. Information theory and neural nets. In *Mathematical Perspectives on Neural Networks*, P. Smolensky, M. Mozer, and D. Rumelhart, Eds. Lawrence Erlbaum Associates, 1996, ch. 16.

[113] Rissanen, J. Oppivien ja älykkäiden järjestelmien sovellukset 1994–1998, tutkimusprojektien arviointi. Tech. Rep. Teknologiaohjelmaraportti 2/97, Teknologian kehittämiskeskus (TEKES), 1997.

[114] Robinson, R. Counting unlabeled asyclic graphs. In *Combinatorial Mathematics*, C. Little, Ed., no. 622 in Lecture Notes in Mathematics. Springer-Verlag, 1977.

[115] Roth, D. On the hardness of approximate reasoning. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence* (1993), vol. 1, pp. 613–618.

[116] Santos, E., and Shimony, S. Belief updating by enumerating high-probability independence-based arguments. In *Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence (UAI–94)* (Seattle, Washington, July 1994), R. Lopez de Mantaras and D. Poole, Eds., Morgan Kaufmann Publishers, San Francisco, CA, pp. 506–513.

[117] Saul, L., Jaakkola, T., and Jordan, M. Mean field theory for sigmoid belief networks. *Journal of Artificial Intelligence Research 4* (1996), 61–76.

[118] Schafer, G., Ed. *Probabilistic Expert Systems.* Society for Industrial and Applied mathematics (SIAM), 1996.

[119] Schafer, G., and Pearl, J., Eds. *Readings in Uncertain Reasoning.* Morgan Kaufmann Publishers, San Mateo, CA, 1990.

[120] Schwarz, G. Estimating the dimension of a model. *Annals of Statistics 6* (1978), 461–464.

[121] Scott, D. *Multivariate Density Estimation. Theory, Practice, and Visualization.* John Wiley & Sons, New York, 1992.

[122] Seppänen, M. Bayes-verkon muodostaminen datasta. Master's thesis, Department of Computer Science, University of Helsinki, 1996.

[123] Shachter, R. Probabilistic inference and influence diagrams. *Operations Research 36*, 4 (July-August 1988), 589–604.

[124] Shimony, S. Finding MAPs for belief networks is NP-hard. *Artificial Intelligence 68* (1994), 399–410.

[125] Singh, M., and Valtorta, M. An algorithm for construction of Bayesian network structures from data. In *Uncertainty in Artificial Intelligence, Proceedings of the Ninth Conference* (1993), E. Horvits and F. Jensen, Eds., Morgan Kaufmann Publishers, San Francisco, CA, pp. 259–265.

[126] Smyth, P., Heckerman, D., and Jordan, M. Probabilistic independece networks for hidden Markov probability models. Tech. Rep. A.I. Memo 1565, C.B.C.L Memo 132, M.I.T, February 1996.

[127] Sorsa, M. Kvalitatiiviset bayesverkot. Master's thesis, Department of Computer Science, University of Helsinki, 1996.

[128] Specht, D. Probabilistic neural networks. *Neural Networks 3* (1990), 109–118.

[129] Spiegelhalter, D., Dawid, P., Lauritzen, S., and Cowell, R. Bayesian analysis in expert systems. *Statistical Science 8*, 3 (1993), 219–283.

[130] Spirtes, P., Glymour, C., and Scheines, R., Eds. *Causation, Prediction and Search.* Springer-Verlag, 1993. Out of print. Can be downloaded from 'URL: http://hss.cmu.edu/html/departments/philosophy/TETRAD.BOOK/book.html'.

[131] Stanfill, C., and Waltz, D. Toward memory-based reasoning. *Communications of the ACM 29*, 12 (1986), 1213–1228.

[132] Tikhonov, A. On solving incorrectly posed problems and method of regularization. *Doklady Akademii Nauk USSR 151* (1963), 501–504.

[133] Tirri, H. *Plausible Prediction by Bayesian Inference.* PhD thesis, Report A-1995-1, Department of Computer Science, University of Helsinki, June 1997.

[134] Tirri, H., Kontkanen, P., and Myllymäki, P. A Bayesian framework for case-based reasoning. In *Advances in Case-Based Reasoning*, I. Smith and B. Faltings, Eds., vol. 1168 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag, Berlin Heidelberg, November 1996, pp. 413–427.

[135] Tirri, H., Kontkanen, P., and Myllymäki, P. Probabilistic instance-based learning. In *Machine Learning: Proceedings of the Thirteenth International Conference* (1996), L. Saitta, Ed., Morgan Kaufmann Publishers, pp. 507–515.

[136] Titterington, D., Smith, A., and Makov, U. *Statistical Analysis of Finite Mixture Distributions.* John Wiley & Sons, New York, 1985.

[137] Vapnik, V. *The Nature of Statistical Learning Theory.* Springer-Verlag, New York, 1995.

[138] Walley, P. Measures of uncertainty in expert systems. *Artificial Intelligence 83* (1996), 1–58.

[139] Watson, I., and Marir, F. Case-based reasoning: A review. *The Knowledge Engineering Review 9*, 4 (1994), 327–354.

[140] Wellman, M. Fundamental concepts of qualitative probabilistic networks. *Artificial Intelligence 44*, 3 (August 1990), 257–304.

[141] Whittaker, J. *Graphical Models in Applied Multivariate Statistics.* John Wiley & Sons, 1990.

[142] YANNAKIS, M. Computing the minimal fill-in is NP-complete. *SIAM Journal of Algebraic Discrete Methods 2* (1981), 77–79.

[143] ZHANG, L., AND POOLE, D. Sidestepping the triangulation problem in Bayesian net computations. In *Proceedings of the Eight Annual Conference on Uncertainty in Artificial Intelligence* (Palo Alto, CA, July 1992), Morgan Kaufmann Publishers, pp. 360–367.