Päättely sekamalleissa

Esa Pitkänen

5. huhtikuuta 2002

Älykkäiden järjestelmien tutkimusseminaari
Tietojenkäsittelytieteen laitos
Helsingin yliopisto

1 Johdanto

Täsmällisellä probabilistisella päättelyllä tarkoitetaan Bayes-verkkojen yhteydessä yleensä tavoiteltavan posteriorijakauman laskemista niin, että tuloksen ensimmäinen ja toinen momentti ovat samat kuin oikealla jakaumalla. Tämä määrittely on peräisin Lauritzenilta jonka klikkipuualgoritmi säilyttää kyseiset momentit [Lau92].

Mielivaltaisissa Bayes-verkoissa täsmällisen probabilistisen päättelyn ongelma on osoitettu $\mathcal{NP}$-kovaksi [Coo90]. Ongelmaan on haettu ratkaisua kahdella tavalla. Asettamalla sopivasti rajoituksia käytettävälle mallille voidaan täsmällisiä päättelyalgoritmeja käyttää tehokkaasti. Approksimaalisten menetelmien avulla voidaan ratkaista päättelyongelmia mielivaltaisissa Bayes-verkoissa. Ongelmaksi muodostuu tällöin approksimointialgoritmien konvergoinnin hitaus ja epätarkkuus.

Viime vuosina on kehitetty menetelmiä, jotka yhdistävät täsmällisen ja approksimaalisen päättelyn parhaat puolet. Suosituin lähestymistapa on ollut yhdistää klikkipuualgoritmiin sopivasti approksimointimenetelmiä. Näin on saatu aikaan algoritmeja, jotka sallivat päättelyn ns. *sekamalleissa* (*hybrid models*), joissa jatkuvien ja diskreettien muuttujien esiintymistä yhdessä ei ole rajoitettu.

Tässä kirjoitelmassa tarkastellaan probabilistisen päättelyn menetelmiä rakenteeltaan rajoittamattomissa Bayes-verkoissa. Menetelmistä tarkastellaan sekä approksimaalisia että täsmällisiä algoritmeja. Kirjoitelmassa esitellään algoritmi täsmälliseen päättelyyn sekamalleissa, joissa diskreettien muuttujien jatkuva-arvoiset lapset mallinnetaan *softmax*-funktioilla [LSK01].

Seuraavassa kappaleessa esitellään Bayes-verkkojen perusteet. Kappaleessa 3 käydään läpi approksimaalisia päättelymenetelmiä. Kappaleessa 4 tarkastellaan täsmällisen päättelyn klikkipuualgoritmia ja sen johdannaisia.

2 Bayes-verkot

Olkoon $X = X_1, X_2, \ldots, X_n$ joukko satunnaismuuttujia. Bayes-verkko on tilataloudellinen tapa esittää muuttujien X yhteistodennäköisyysjakauma olettamalla joukko muuttujien välisiä riippumattomuuksia ja koodaamalla oletusten perusteella yhteisjakauma joukoksi ehdollisia todennäköisyysjakaumia [Pea88].

Bayes-verkko G muuttujille X muodostuu kvalitatiivisesta ja kvantitatiivisesta osasta. Kvalitatiivinen osa on suunnattu verkko (V, E), jossa V on satunnaismuuttujiin X samaistuva joukko solmuja ja E on kaarten joukko. Kaaret E koodaavat muuttujien väliset itsenäisyysoletukset.

Muuttujiin X liittyvät ehdolliset todennäköisyysjakaumat muodostavat Bayes-verkon kvantitatiivisen osan siten, että jokaiselle muuttujalle X_i on kiinnitetty jakauma $P(X_i \mid X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n)$. Verkon kaarten E koodaamien itsenäisyysoletusten perusteella tämä voidaan kirjoittaa muotoon

$$P(X_i \mid X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n) = P(X_i \mid Pa(X_i)),$$

missä muuttujan X_i vanhempien joukkoa merkitään $Pa(X_i)$.

Yhteistodennäköisyysjakauma $P(X_1, X_2, \ldots, X_n)$ voidaan todennäköisyyksien ketjusäännön mukaan kirjoittaa muotoon

$$P(X_1, X_2, \ldots, X_n) = \prod_{i=1}^{n} P(X_i \mid Pa(X_i).$$

Probabilistisella päättelyllä tarkoitetaan tässä kiinnostavien marginaalitodennäköisyysjakaumien ja maksimitodennäköisyyskonfiguraatioiden (maximum a posteriori, MAP) laskemista. Muuttujien arvoista saattaa olla saatavilla havaintoja, jotka halutaan saada mukaan marginaalien laskentaan.

Täsmällinen päättely mielivaltaisissa Bayes-verkoissa on $\mathcal{NP}$-kova ongelma [Coo90]. Myös approksimaalisen päättelyn ongelma mielivaltaisissa verkoissa on $\mathcal{NP}$-kova [DL93]. Tämän vuoksi käytettävän mallin rakenteelle asetetaan yleensä rajoituksia, jotta päättely olisi käytännössä mahdollista. Erityisesti yksipolkuisissa, vain diskreettejä muuttujia sisältävissä puissa (polytrees) päättelyn aikavaativuus on lineaarinen suhteessa muuttujien lukumäärään. Täsmällistä ja approksimaalista päättelyä ei kuitenkaan voida $\mathcal{NP}$-kovuudesta huolimatta voida pitää “yhtä vaikeina” ongelmina. Useissa tapauksissa approksimoimalla on löydetty kelvollinen ratkaisu päättelyongelmaan, jota ei ole voitu ratkaista täsmällisin menetelmin [JJ96].

Yleisesti käytettävä Bayes-verkkoluokka, jossa sallitaan sekä diskreetit että jatkuvat muuttujat, on *ehdollisten lineaaristen normaalijakaumien* (*Conditional Linear Gaussian*, CLG) verkkoluokka [LSK01]. CLG-mallissa ei sallita jatkuva-arvoisille muuttujille diskreettejä lapsia. Diskreettien muuttujien jakaumat annetaan todennäköisyystauluina, jatkuva-arvoiset muuttujat mallinnetaan ehdollisilla lineaarisilla normaalijakaumilla. Jokaisen jatkuva-arvoisen muuttujan diskreettien vanhempien arvoyhdistelmää kohden määritetään oma normaalijakaumansa, joka on muotoa

$$P(X \mid y, d) = N(a_{d,0} + \sum_{i=1}^{k} a_{d,i} y_i; \sigma_d^2),$$

missä a ja σ^2 ovat normaalijakauman parametrit, d on diskreettien ja y jatkuva-arvoisten vanhempien arvoyhdistelmä. Normaalijakauma ei luonnollisesti sovi kaikkien jatkuva-arvoisten muuttujien mallintamiseen, mutta sen käyttö on yleistä päättelyssä tarvittavien laskutoimitusten yksinkertaisuuden vuoksi.

Päättely CLG-mallissa on kuitenkin $\mathcal{NP}$-kova ongelma. Tämä pätee myös yksipolkuisissa CLG-malleissa, joissa jokaisella jatkuvalla muuttujalla on enintään yksi binäärinen diskreetti esi-isä [LP01]. Edes approksimaalinen päättely ei ole tehokasta näinkin yksinkertaisissa malleissa. Diskreettien lasten kieltäminen jatkuvilta vanhemmilta on rajoitus, joka estää monien luonnollisten tapausten mallintamisen Bayes-verkoilla. Esimerkiksi luonnossa jatkuva-arvoisesta suureesta saatetaan saada tietoja vain diskreettejä tuloksia antavan havaintolaitteen avulla. Tällöin ei voida rajoittua CLG-malleihin.

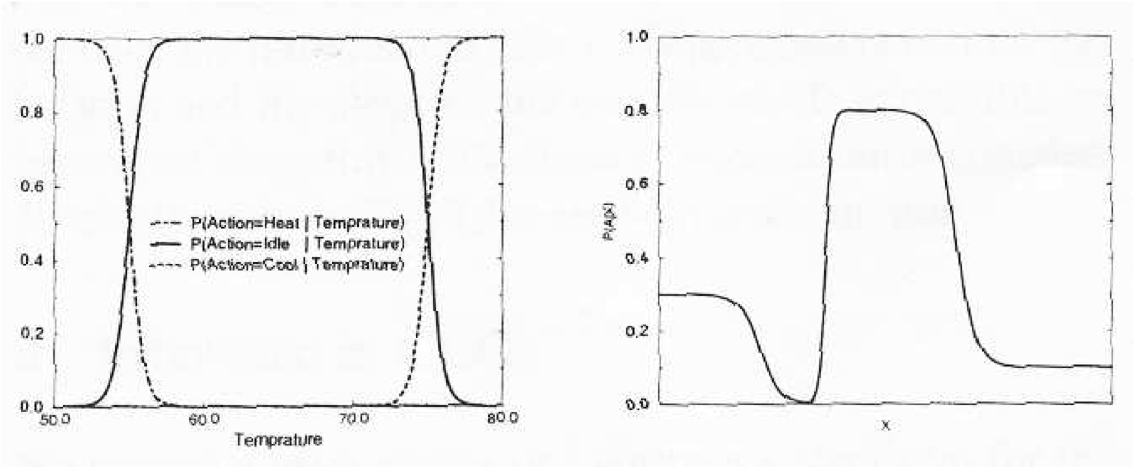
Yleinen tapa laajentaa CLG-mallia sallien diskreeteille lapsille jatkuva-arvoiset vanhemmat, on mallintaa diskreetit muuttujat *softmax*-funktioina, joita kutsutaan myös *logistisiksi* funktioiksi [LSK01]. Tällöin diskreetin muuttujan A todennäköisyysjakauma ehdolla sen vanhemmat $Y_1, \ldots, Y_k$ on

$$P(A = a_i \mid y_1, \ldots, y_k) = \frac{\exp\left(b^i + \sum_{l=1}^{k} w_l^i y_l\right)}{\sum_{j=1}^{m} \exp\left(b^j + \sum_{l=1}^{k} w_l^j y_l\right)}.$$

Muuttujille, joilla on diskreettejä vanhempia, määritellään yksi softmax-jakauma jokaista diskreettien vanhempien arvoyhdistelmää kohden. Kuvassa 1 on kaksi softmax-kuvaajaa. Riippuen jatkuvien muuttujien arvoista, yksi muuttujan tilatodennäköisyyksistä kerrallaan on vallitseva.

Tarkastellaan seuraavassa erilaisia tapoja suorittaa probabilistista päättelyä Bayes-verkossa, jossa jatkuvilla muuttujilla sallitaan olevan diskreettejä lapsia. Täsmällinen päättely yhdistetään nykyisin lähes aina Lauritzenin klikkipuualgoritmiin, joka esitellään luvussa 4.1. Klikkipuualgoritmin käsittelyn jälkeen esitellään kolme menetelmää, jotka perustuvat klikkipuualgoritmin välitulosten approksimoimiseen: epätasainen diskretointi, variationaalinen approksimointi ja LSK-algoritmi.

Tämän kirjoitelman ulkopuolelle jääviä päättelymenetelmiä ovat mm. *bucket elimination* [Dec98] ja *symbolic probabilistic inference* [CF91, CF95]. Rajaus ei ole tyydyttävästi pe-



Kuva 1: Softmax-funktioita

rusteltavissa, vaan perustuu kirjoitelman laadinnassa suoritettuun artikkelien stokastiseen otantaan. Jälkimmäisen menetelmän tapauksessa kilpailevan algoritmin kehittäjät väittävät marginaalien laskemisen vaativan $\frac{N}{2}$ pyyhkäisyä (passes) siinä missä klikkipuualgoritmi vaatii 2 pyyhkäisyä.

3 Stokastiset otantamenetelmät

Otantamenetelmissä (*sampling*) haluttua posteriorijakaumaa arvioidaan Bayes-verkosta generoitujen otosten perusteella. Toisin kuin täsmällisen päättelyn menetelmissä, verkon topologia ei vaikuta laskennan vaativuuteen. Otantamenetelmät ovat iteratiivisia: tuloksen tarkkuus ja aikavaatimus ovat suoraan verrannollisia otosten lukumäärään. Laskenta voidaan keskeyttää milloin tahansa, jolloin päättelyyn käytettävissä olevat resurssit voidaan hyödyntää tarkasti.

Otantapäättelyalgoritmien suurimmat puutteet ovat konvergoinnin hitaus ja ongelmat *epätodennäköisten havaintojen* (*unlikely evidence*) käsittelyssä. Suurissa Bayes-verkoissa kohtuulliseen tarkkuuteen vaadittavien otosten määrä kasvaa usein käytännössä liian suureksi. Tällöin on tyydyttävä pienempään tarkkuuteen tai, verkon topologian niin salliessa, käytettävä täsmällisiä menetelmiä[1]. Eräissä tapauksissa voidaan otosavaruuden kokoa rajoittaa ottamalla huomioon mallin erityispiirteet. Esimerkiksi vikadiagnostiikkamalleissa voidaan keskittyä hypoteeseihin, joissa esiintyy vähän vikatilanteita kerrallaan [LP01].

Tässä kirjoitelmassa tarkasteltavat otantamenetelmät ovat Gibbsin otanta ja painotettu otanta.

[1]Joita tosin varmaankin käytetään, jos se vain on mahdollista.

3.1 Gibbsin otanta

Gibbsin otanta (*Gibbs sampling*) on yksinkertainen *Markovin ketju Monte Carlo* -menetelmä (*Markov chain Monte Carlo*, *MCMC*) joka on erikoistapaus yleisemmästä Metropolis-algoritmista [Nea93]. Muiden Monte Carlo -menetelmien tavoin Gibbsin otanta perustuu kohdefunktion $a(\mathbf{X})$ odotusarvon $E[a]$ arvioimiseen N otoksen avulla,

$$E[a] = \sum_{\mathbf{x}} a(\mathbf{x}) P(\mathbf{X} = \mathbf{x}) \approx \frac{1}{N} \sum_{t=1}^{N} a(\mathbf{x}^{(t)}),$$

missä $\mathbf{X} = \{X_1, \ldots, X_n\}$ on joukko satunnaismuuttujia, $\mathbf{x} = x_1, \ldots, x_n$ ovat satunnaismuuttujien mahdolliset arvot ja $\mathbf{x}^{(t)}$ on satunnaismuuttujien arvoyhdistelmä otoksessa t.

Yleisesti ottaen Gibbsin otanta on käyttökelpoinen ratkaistaessa ongelmia, joissa satunnaismuuttujien arvoalueet ovat äärellisiä ja pieniä tai muuttujilla on ehdolliset todennäköisyysjakaumat, joista otanta on helppoa. Tämän vuoksi Gibbsin otannan käyttö Bayes-verkossa, jonka kaikki parametrit tunnetaan, on helppoa: jokaisella solmulla on kiinnitetty todennäköisyysjakauma ehdolla solmun vanhemmat.

Gibbsin otannassa otos $\mathbf{X}^{(t)}$ saadaan otoksesta $\mathbf{X}^{(t-1)}$ käymällä muuttujia läpi korvaten yksi kerrallaan jokaisen muuttujan X_i arvo ehdollisesta jakaumasta
$P(X_i \mid X_1, \ldots, X_{i-1}, X_{i+1}, \ldots, X_n)$ otetulla arvolla. Muuttujat käydään yleensä läpi järjestyksessä, mutta myös satunnaisella läpikäyntijärjestyksellä voidaan algoritmille osoittaa samat ominaisuudet. Bayes-verkossa Gibbsin otanta voi hyödyntää verkon koodaamia muuttujien riippumattomuusoletuksia ja valita arvot jakaumasta $P(X_i \mid Pa(X_i))$. Tarkemmin otosten arvojen valintaa voidaan kuvata seuraavasti.

$$\begin{array}{l}
\text{Valitaan muuttujan } X_1^{(t)} \text{ arvo jakaumasta } P(X_1 \mid x_2^{(t-1)}, x_3^{(t-1)}, \ldots, x_n^{(t-1)}). \\
\text{Valitaan muuttujan } X_2^{(t)} \text{ arvo jakaumasta } P(X_2 \mid x_1^{(t)}, x_3^{(t-1)}, x_4^{(t-1)}, \ldots, x_n^{(t-1)}). \\
\vdots \\
\text{Valitaan muuttujan } X_i^{(t)} \text{ arvo jakaumasta } P(X_i \mid x_1^{(t)}, \ldots, x_{i-1}^{(t)}, x_{i+1}^{(t-1)}, \ldots, x_n^{(t-1)}). \\
\vdots \\
\text{Valitaan muuttujan } X_n^{(t)} \text{ arvo jakaumasta } P(X_n \mid x_1^{(t)}, \ldots, x_{n-1}^{(t)}).
\end{array}$$

Näin edetään, kunnes N otoksen arvot on saatu selville. Otanta ohjautuu jakaumassa alueille, joiden todennäköisyysmassa on suuri. Vaikka otokset ovatkin jossain mielessä riippuvia toisistaan, voidaan Markovin ketjujen teorian perusteella sanoa otannan olevan "reilua" ainakin asymptoottisesti.

Voidaan osoittaa, että Gibbsin otannalla saatava tulos lähestyy oikeaa arvioitavan funktion arvoa, kun otosten määrä N lähestyy ääretöntä. Käytännössä ratkaistaessa päättelyongelmia Bayes-verkoissa saattaa Gibbsin otannan konvergointi olla liian hidasta. Menetelmän yleisestä luonteesta johtuen se on kuitenkin käytettävissä mielivaltaisissa Bayes-verkoissa:

ehdollisten todennäköisyysjakaumien muotoa ei rajoiteta millään tavalla, vaan ne voidaan esittää todennäköisyystauluina tai tiheysfunktioina. Näin ollen menetelmää voidaan käyttää päättelyyn myös sekamalleissa, jotka ovat pitkään olleet täsmällisten menetelmien ulottumattomissa.

3.2 Painotettu otanta

Painotetussa otannassa (*importance sampling*) generoidaan riippumattomia otoksia vaikeasti saatavilla olevan jakauman P sijaan muunnellusta jakaumasta Q [SP90]. Jakauma Q saadaan muistuttamaan jakaumaa P painottamalla otoksia sopivasti. Tarkemmin sanoen otoksiin $x_1, \ldots, x_t$ liitetään paino $w(x_i)$ siten, että $w(x_i) = P(x_i)/Q(x_i)$. Tällöin oletetaan, että $P(x)$ pystytään laskemaan yksittäisissä pisteissä.

Bayes-verkossa $P(x)$ ei kuitenkaan ole helposti laskettavissa. Yleisesti käytetty ratkaisu ongelmaan on nimeltään *epäsuora painotettu otanta* (*indirect importance sampling*, *likelihood weighting*) [SP90, FC90]. Siinä otosten painot ovat muotoa $u(x_i) = \beta\frac{P(x_i)}{Q(x_i)}$, missä β on kiinnitetty vakiokerroin, jota ei tarvitse tuntea etukäteen eikä myöskään laskea otannan aikana. Jakaumaehdotus Q ja painot voidaan Bayes-verkossa valita sopivasti tilanteen mukaan. Lopuksi tavoiteltu jakauma P approksimoidaan otoksista Monte Carlo -approksimoinnilla, kuten Gibbsin otannassakin.

Valitettavasti painotettua otantaa vaivaavat samat ongelmat kuin Gibbsin otantaakin. Approksimoitu jakauma lähestyy oikeaa jakaumaa vasta, kun otosten määrä lähestyy ääretöntä. Konvergointi käytännön sovelluksissa on osoittaunut hitaaksi varsinkin, kun muuttujia on paljon. Lisäksi menetelmän konvergointi hidastuu merkittävästi tapauksissa, joissa havainnot ovat epätodennäköisiä. Myös painotettu otanta on sovellettavissa kaikkiin Bayes-verkkoihin riippumatta valituista todennäköisyysmalleista tai verkon topologiasta.

Painotettu otanta on ryhmä menetelmiä, joihin kuuluu epäsuoran painotetun otannan lisäksi esimerkiksi *probabilistic logic sampling* [Hen86] ja *stratified simulation* [Bou94]. Näistä logic sampling on ensimmäinen ja myös yksinkertaisin. Epäsuora otanta on suosittu mm. vertailtaessa uusien sekamallien päättelymenetelmien suorituskykyä. Tämän voi tulkita tarkoittavan sitä, että menetelmää pidetään hyvänä.[2]

4 Klikkipuualgoritmi ja johdannaiset

Seuraavaksi esitellään vallitseva täsmällisen päättelyn algoritmi, klikkipuualgoritmi. Algoritmille on esitetty useita laajennoksia ja monet approksimointimenetelmät perustuvat siihen, esimerkkinä HUGS, jossa klikkipuualgoritmin välituloksia approksimoidaan Gibbsin otannalla [KjĄ95] Kappaleessa 4.2 kuvataan lyhyesti epätasaiseen diskretointiin perus-

[2]Tai sitten sitä vasten vertailtaessa vain saadaan oma algoritmi näyttämään kelvolliselta.

tuva päättely, kappaleessa 4.3 variationaalista approksimointia hyödyntävä menetelmä ja lopuksi kappaleessa 4.4 menetelmä täsmälliseen päättelyyn sekamalleissa.

4.1 Klikkipuualgoritmi

Klikkipuualgoritmi (*clique tree algorithm*, *join tree algorithm*, *probability propagation in trees of clusters, PPTC*) on yleisimmin käytetty Bayes-verkkojen probabilistisen päättelyn algoritmi. Siinä Bayes-verkko muunnetaan aluksi yksipolkuiseksi rakenteeksi, *klikkipuuksi* (*join tree, clique tree, junction tree, cluster tree*), jossa päättely suoritetaan. Algoritmi on täsmällinen siinä suhteessa, että se laskee tavoitejakauman $P(x \mid e)$ oikein ensimmäisen ja toisen momentin tarkkuudella. Algoritmin kuvaus yksityiskohtaisella tasolla olisi ollut tässä kirjoitelmassa esitettäväksi liian pitkä: menetelmästä julkaistut paperit ([Lau92, LJ01]) tarjoavat kattavan selvityksen. Murphyn esitys on käytännönläheisempi [Mur98]. Seuraavassa käydään algoritmi läpi yleistasolla.

Algoritmi perustuu ajatukseen, että kohdejakauman laskemisesta voidaan selvitä sarjalla paikallisia laskutoimituksia. Paikallisella tarkoitetaan Bayes-verkoissa joukkoa muuttujia, jotka sijaitsevat sopivasti toistensa läheisyydessä muodostaen klikkipuussa klikin, eli täysin kytketyn solmujoukon, joka ei ole suuremman täysin kytketyn solmujoukon osajoukko. Marginaalitodennäköisyyksien laskenta palautuu klikkipuussa klikkikohtaisten jakaumien käsittelyksi ja klikkienväliseksi viestien vaihdoksi.

Ennen varsinaisen klikkipuun muodostamista Bayes-verkko moralisoidaan ja kolmioidaan. Moralisoinnissa Bayes-verkko muutetaan suuntaamattomaksi ja kahden muuttujan välille lisätään suuntaamaton kaari, jos ja vain jos niillä on yhteinen lapsi. Kolmioinnissa yli kolmen muuttujan muodostamiin kaariin lisätään kaaria, kunnes kaaria ei enää voida lisätä. Optimaalinen kolmiointi on $\mathcal{NP}$-täydellinen ongelma. Ratkaisun ei kuitenkaan tarvitse olla optimaalinen; kolmiointi voidaan ratkaista polynomisessa ajassa heuristisella menetelmällä.

Kolmioinnissa saatu verkko voidaan nyt muuntaa klikkipuuksi. Aluksi verkosta etsitään kaikki klikit. Klikit sopivasti yhdistämällä saadaan yksipolkuinen, suuntaamaton klikkipuu (kts. esim. [HD96]). Jokaiseen saatuun klikkipuun solmuun v (klikkiin) yhdistetään potentiaalifunktio ϕ_v. Potentiaalifunktio on kuvaus solmuun liitetyiltä muuttujilta reaalilukuun. Aluksi potentiaalifunktio ϕ_v koostuu ehdollisten todennäköisyysfunktioiden tulosta. Jokainen funktioista on määritelty Bayes-verkon muuttujalle ja tämän vanhemmille siten, että nämä kaikki kuuluvat samaan klikkipuun solmuun v.

Klikkipuussa vierekkäisten solmujen A ja B välille lisätään *erottajasolmu* (*separator*) S, johon liitetään solmujen A ja B yhteisten muuttujien joukko ja näitä vastaavat potentiaalifunktiot. Klikkipuun solmun (tavallinen tai erottaja) potentiaalifunktio esittää yhteistodennäköisyysjakaumaa yli solmun muuttujien. Pelkästään diskreettejä muuttujia sisältävässä solmussa voidaan käyttää moniulotteisia taulukoita potentiaalin kuvaamiseen.

Tässä rakenteessa yhteisjakauma $P(U)$ yli kaikkien mallin muuttujien on

$$P(U) = \frac{\prod_{V \in \mathcal{C}} \phi_V}{\prod_{S \in \mathcal{S}} \phi_S}, \tag{1}$$

missä $\mathcal{C}$ on solmujen joukko ja $\mathcal{S}$ on erottajien joukko [Lau92].

Havainnot muuttujien tiloista otetaan huomioon seuraavasti. Oletetaan, että yhden muuttujan A tila on tiedossa, $A = a$. Uskomusten päivittäminen aloitetaan tällöin klikkipuun solmusta, joka sisältää muuttujan A ja tämän vanhemmat $Pa(A)$. Solmun potentiaaliin asetetaan havainnon $A = a$ kanssa ristiriidassa olevien komponenttien todennäköisyydeksi nolla ja potentiaali normalisoidaan.

Tämän jälkeen potentiaalin muutos välitetään klikkipuun muihin solmuihin (*propagation*). Havaintojen välittäminen klikkipuuhun tehdään kahdessa vaiheessa vaihtamalla viestejä solmun välillä. Aluksi valitaan jokin klikeistä juurisolmuksi R. Ensimmäisessä vaiheessa viestit kulkevat juuresta kohti lehtiä. Vaihetta kutsutaan havaintojen keräämiseksi (*collect evidence phase*). Tämän jälkeen havainnot levitetään (*distribute evidence phase*) verkkoon välittämällä viestejä puun lehdistä juurta kohden. Algoritmin päätyttyä yhtälö (1) on voimassa ja minkä tahansa muuttujan marginaalitodennäköisyys voidaan laskea.

Klikkipuualgoritmin aikavaativuus on polynomiaalinen suhteessa klikkipotentiaalien kokoihin. Tämä puolestaan kasvaa eksponentiaalisesti diskreettien muuttujien lukumäärän suhteen ja neljänteen potenssiin jatkuvien muuttujien lukumäärän suhteen. Näin ollen juuri diskreetit muuttujat aiheuttavat algoritmille ongelmia.

Suurempia ongelmia tuottaa laskennan klikkikohtaisien välitulosten käsittely. Joissakin suurissa Bayes-verkoissa klikkipuualgoritmin käyttö on käytännössä mahdotonta. Jaakkola ja Jordan esittelevät tapauksen, jossa tavanomaisen klikkipuualgoritmin käyttö on mahdotonta [JJ96]. Kyseessä on lääketieteellinen hyvin tiheästi kytketty malli, jossa klikkikoot ovat sadan muuttujan luokkaa: päättely mallissa onkin perinteisesti järjestetty erilaisin approksimointimenetelmin.

4.2 Diskretointiin perustuva päättely

Mielivaltainen, sekä jatkuva-arvoisia että diskreettejä muuttujia sisältävä Bayes-verkko voidaan diskretisoinnin avulla palauttaa täysin diskreetiksi malliksi. Diskretointi voidaan esimerkiksi tehdä yksinkertaisesti jakamalla jokaisen jatkuva-arvoisen muuttujan arvoalue tasaisesti osaväleihin (*uniform discretization*). Tällöin päättelyyn voidaan käyttää esimerkiksi klikkipuualgoritmia. Diskretointi on käytännössä ollut hyvin yleinen keino selviytyä päättelystä sekamalleissa.

Lähestymistavasta seuraa kuitenkin vaikeuksia. Jos diskretoidussa mallissa päättely suoritetaan klikkipuualgoritmilla, on kiinnitettävä huomiota syntyviin potentiaalien kokoihin. Käytännössä haluttuun tarkkuuteen tasaisessa diskretoinnissa ei useinkaan päästä, sillä potentiaalit kasvavat helposti liian isoiksi. Lisäksi kelvollisen diskretoinnin löytäminen on usein epätriviaalia ja siten hidasta varsinkin, jos ulottuvuuksien lukumäärä on suuri [KK97].

Tasainen diskretointi käyttää yleensä paljon tilaa jakaumien "tasaisten", epäinformatiivisten kohtien esittämiseen, joten luonnollinen ajatus on pyrkiä diskretointiin, joka kadottaa mahdollisimman vähän informaatiota. Parempi diskretointi saadaankin aikaan *epätasaisella dynaamisella diskretoinnilla* (*nonuniform dynamic discretization*), jossa ositetaan jokaisen klikin potentiaalifunktio muuttujien ehdollisen jakaumien sijaan [KK97]. Epätasainen diskretointi voidaan yhdistää luontevasti klikkipuualgoritmiin. Yhdistettyä algoritmia joudutaan iteroimaan, mutta sen voidaan todistaa konvergoituvan. Menetelmän kehittäjien kokeissa algoritmi konvergoi nopeasti, useimmissa tapauksissa toisella iteraatiolla.

Epätasainen diskretointi on herkkä epätodennäköisten havaintojen vaikutuksille. Osituksen muuttaminen päättelyn aikana havaintojen perusteella parantaa kuitenkin tarkkuutta.

4.3 Variationaalinen approksimointi

Murphy [Mur99] kuvaa *variationaalisia menetelmiä* (*variational methods*) hyödyntävän päättelyalgoritmin sekamalleille, joissa jatkuvien vanhempien diskreetit lapset mallinnetaan softmax-funktioilla. Ajatuksena on muuttaa softmax-funktiot normaalijakaumiksi, jolloin klikkipuualgoritmissa suoritettavat marginalisoinnit voidaan laskea suljetussa muodossa. Tämä ajatus on myös myöhemmin esiteltävän LSK-algoritmin perustana.

Wiegerinckin menetelmässä [Wie00] käytetään fysiikasta peräisin olevaa mean field -approksimointia, joka kuuluu variationaalisiin menetelmiin. Olkoon $P(x)$ todennäköisyysjakauma yli diskreettien muuttujien $x = x_1, \ldots, x_n$, jotka saavat äärellisiä arvoja, $x_i \in \{1, \ldots, n_i\}$. Oletetaan, että P voidaan jakaa osiin,

$$P(x) = \frac{1}{Z_P} \prod_\alpha \Psi_\alpha(d_\alpha),$$

missä Ψ_α ovat pienestä muuttujajoukosta d_α riippuvia potentiaalifunktioita ja Z_P on normalisointitekijä [Wie00]. Bayes-verkon koodaama yhteistodennäköisyysjakauma ehdolla havainnot $P_e(x)$ on tätä muotoa,

$$P_e(x) = P(x \mid e) = \frac{\prod_j P(x_j \mid \pi_j)}{P(e)},$$

jolloin $\Psi_j(d_j) = P(x_j \mid \pi_j)$, $d_j = (x_j, \pi_j)$ ja $Z_P = P(e)$. Klikkipuualgoritmin käsittelemät klikkipotentiaalit ovat sopivasti samaa muotoa. Tavoiteltavaa jakaumaa $P(x)$ arvioidaan vastaavasti tulomuodossa kirjoitettavalla jakaumalla $Q(x)$,

$$Q(x) = \frac{1}{Z_Q} \prod_{\gamma} \Phi_\gamma(d_\gamma).$$

Approksimointia parannetaan minimoimalla jakaumien P ja Q välistä Kullback-Leibler -divergenssiä $D(Q, P)$,

$$D(Q, P) = \sum_{\{x\}} Q(x) \log \frac{Q(x)}{P(x)},$$

jonka voidaan tässä tapauksessa todistaa etsiytyvän aina paikalliseen minimiin. Ongelmaksi jää vielä jakauman Q valinta. Eräs tapa on käyttää potentiaaleja P ja vain korvata potentiaalien vaikeasti laskettavat tapaukset variationaalisilla parametreilla.

Murphyn menetelmä pärjäsi hyvin vertailussa Gibbsin otantaa vastaan sekä nopeutensa että tarkkuutensa puolesta [Mur99]. Wiegerinck ei tarjoa empiirisiä tuloksia menetelmänsä käyttökelpoisuudesta. Molemmissa menetelmissä diskreetit muuttujat ovat binäärisiä, eikä suorituskyky- tai tarkkuustakeita anneta.

Sopivan variationaalisen approksimoinnin avulla päättely on mahdollista myös jossain tiheissä malleissa, kuten QMR-DT -mallissa [JJ96]. Variationaalinen menetelmä on merkitsevästi otantamenetelmää parempi niin nopeudessa kuin tarkkuudessakin.

4.4 LSK-algoritmi

Lerner, Segal ja Koller [LSK01] esittävät klikkipuualgoritmin päälle rakentuvan menetelmän, jolla päättelyongelmia voidaan ratkaista täsmällisesti sekamalleissa. Mallin oletetaan olevan laajennettu CLG-malli, jossa jatkuvien muuttujien diskreetit lapset mallinnetaan softmax-funktioilla. Algoritmi laskee tuloksen ensimmäisen ja toisen momentin oikein.

Algoritmin pohjana on Murphyn [Mur99] esittämä normaali- ja softmax-jakauman tulon approksimointi normaalijakaumaksi. Tämä approksimointi on välttämätön, sillä klikkipuualgoritmin välittämät viestit ovat klikkikohtaisesti laskettuja normaalijakaumina esitettyjä marginaaleja. Algoritmissa vaaditaan, että moralisoidulle Bayes-verkolle suoritetaan *vahva kolmiointi* (*strong triangulation*). Vahvalla kolmioinnilla tarkoitetaan tilannetta, jossa muuttujan X kaikki diskreetit naapurit sisältyvät samaan klikkiin. Ominaisuus vaaditaan, jotta marginalisointi yli klikin diskreettien muuttujien olisi mahdollista. Kyseinen operaatio vaaditaan klikistä lähetettävän viestin konstruoinnissa.

LSK-algoritmi. Laskettava todennäköisyysjakauma on $P(T, e)$, missä T on kysely (joukko muuttujia) ja e ovat havainnot muuttujien arvoista.

1. Muodostetaan vahvasti kolmioitu klikkipuu.

2. Lisätään potentiaaleihin kaikki ehdolliset jakaumat lukuunottamatta softmax-jakaumia.

3. Kalibroidaan puu klikkipuualgoritmilla.

4. Lisätään havainnot jatkuva-arvoisista muuttujista ja kalibroidaan puu uudelleen.

5. Alustetaan ehdolliset jakaumat havainnoilla jatkuva-arvoisista muuttujista.

6. Jokaista maksimaalista yhdistettyä komponenttia X_i kohden

   (a) Etsitään kaikki softmax-jakaumat $S1, \ldots, S_n$, jotka voidaan lisätä klikkiin C_i ja

   (b) lisätään $S1, \ldots, S_n$ klikkiin C_i.

7. Kalibroidaan puu uudelleen.

8. Palautetaan T:n jakauma.

Ehdollinen jakauma $P(V \mid Pa(V))$ lisätään potentiaaliin yksinkertaisesti kertomalla potentiaali jakaumalla, $\phi_i := \phi_i P(V \mid Pa(V))$. Puun kalibroimisella tarkoitetaan tässä klikkipuualgoritmin ajamista niin, että laskennan päätyttyä yhtälö (1) on voimassa. Vaiheessa 2 ei vielä lisätä softmax-jakaumia, sillä tarvittavaa laskutoimitusta ei voida toteuttaa rakenteessa, jossa yhtälö (1) ei ole voimassa. Tämän vuoksi joudutaan ajamaan klikkipuualgoritmi yhtälön saattamiseksi voimaan. Samalla kierretään tekninen ongelma, joka muuten aiheuttaisi jonkin verran virhettä tulokseen. Lerner, Segal ja Koller mainitsevat myös kaksi muuta luonteeltaan teknistä seikkaa, joiden huomiotta jättäminen tekisi algoritmista approksimaalisen täsmällisen sijaan. Näitä yksityiskohtia ei kuitenkaan käsitellä tässä kirjoitelmassa.

Kehittäjiensä kokeissa LSK-algoritmi oli merkittävästi nopeampi kuin Gibbsin otantaan ja painotettuun otantaan perustuvat algoritmit. Tuloksissa ei ole mitään yllättävää, sillä algoritmin nopeuden ja tarkkuuden on todistettu olevan samaa luokkaa kuin klikkipuualgoritmilla. Mallit, joissa kokeet suoritettiin, olivat klikkipuualgoritmille soveltuvia pieniä, usein kirjallisuudessa esiintyviä Bayes-verkkoja.

5 Yhteenveto

Probabilistiseen päättelyyn tuskin löydetään ongelman $\mathcal{NP}$-kovuuden vuoksi mitään menetelmää, joka olisi ylivoimainen kaikissa malleissa. Sen sijaan on luultavaa, että erilaisiin tapauksiin sovitettujen algoritmien kehitys jatkuu. Jos halutaan käyttää sekamalleja, on joko rajoitettava mallin rakennetta tai tyydyttävä approksimaalisiin, mahdollisesti hitaasti konvergoiviin menetelmiin. Erityisesti tiheissä malleissa täsmällisen ratkaisun laskeminen ei ole suoraviivaista klikkipuualgoritmin avulla liian suuriksi kasvavien klikkikokojen vuoksi. Tällöin on ehkä mahdollista löytää sopiva approksimaatio mallin vaikeasti laskettaville kohdille ja käsitellä mallin loppuosa täsmällisesti. Klikkipuualgoritmin selvä etu verrattuna otantamenetelmiin on Bayes-verkon rakenteellisten ominaisuuksien hyödyntäminen.

Algoritmi jakaa päättelyongelman klikkikohtaisiin laskutoimituksiin ja välitulosten välittämiseen rakenteessa.

Toinen tässä esitettyjen klikkipuualgoritmille perustuvien menetelmien ongelma on niiden rajoittuminen tiettyihin todennäköisyysjakaumiin muuttujia mallinnettaessa. Normaali- ja softmax-jakaumat ovat suorastaan yleishyödyllisiä, mutta ne eivät silti sovi kaikenlaisten ilmiöiden mallintamiseen. Approksimointimenetelmillä saadaan ratkaisu myös erikoisempia jakaumia sisältävissä malleissa. Klikkipuualgoritmissa voidaan hyödyntää paikallisuutta ja laskea täsmällisesti ne klikit, joiden muuttujien jakaumat sen sallivat ja approksimoida loput.

Lähteet

[Bou94] R. R. Bouckaert. A stratified simulation scheme for inference in Bayesian belief networks. In *Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence (UAI'94)*, pages 110–117, July 1994.

[CF91] K. C. Chang and R. M. Fung. Symbolic probabilistic inference with continuous variables. In *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence*, pages 77–85, 1991.

[CF95] K. C. Chang and R. M. Fung. Symbolic probabilistic inference with both discrete and continuous variables. *IEEE Trans. on Systems, Man, and Cybernetics*, 6(25):910–917, 1995.

[Coo90] G. F. Cooper. The computational complexity of probabilistic reasoning using Bayesian belief networks. *Artificial Intelligence*, 42(2-3):393–405, 1990.

[Dec98] R. Dechter. *Bucket elimination: A unifying framework for probabilistic inference. In: Learning in Graphical Models, pages 211–219.* MIT Press, 1998.

[DL93] P. Dagum and M. Luby. Approximating probabilistic inference in Bayesian belief networks is NP-hard. *Artificial Intelligence*, 1(60):141–153, 1993.

[FC90] R. Fung and K. C. Chang. Weighing and integrating evidence for stochastic simulation in Bayesian networks. In M. Henrion, R. Shachter, I. Kanal, and J. Lemmer, editors, *Uncertainty in Artificial Intelligence 5.* Amsterdam: Elseview, 1990.

[HD96] C. Huang and A. Darwiche. Inference in belief networks: A procedural guide. *International Journal of Approximate Reasoning*, 15(3):225–263, 1996.

[Hen86] M. Henrion. Propagating uncertainty by logic sampling in Bayes' networks. In *Proceedings of the Second Workshop on Uncertainty in Artificial Intelligence*, 1986.

[JJ96] T. Jaakkola and M. Jordan. Computing upper and lower bound on likelihoods in intractable networks. In *Proceedings of the Twelfth Annual Conference on Uncertainty in Artificial Intelligence (UAI'96)*, pages 340–348, 1996.

[KjĄ95] U. KjĄærulff. HUGS: Combining exact inference and Gibbs sampling in junction trees. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 1995.

[KK97] A. Kozlov and D. Koller. Nonuniform dynamic discretization in hybrid networks. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, pages 314–325, August 1997.

[Lau92] S. L. Lauritzen. Propagation of probabilities, means and variances in mixed graphical association models. *Journal of the American Statistical Association*, (87):1098–1108, 1992.

[LJ01] S. L. Lauritzen and F. Jensen. Stable local computation with conditional Gaussian distributions. *Statistics and Computing*, (11):191–203, 2001.

[LP01] U. Lerner and D. Parr. Inference in hybrid networks: Theoretical limits and practical algorithms. In *Proceedings of the 17th Annual Conference on Uncertainty in Artificial Intelligence (UAI-01) Seattle, Washington*, pages 310–318, August 2001.

[LSK01] U. Lerner, E. Segal, and D. Koller. Exact inference in networks with discrete children of continuous parents. In *Proceedings of the 17th Annual Conference on Uncertainty in Artificial Intelligence (UAI-01) Seattle, Washington*, pages 319–328, August 2001.

[Mur98] K. P. Murphy. Inference and learning in hybrid Bayesian networks. Technical report, Computer Science Division (EECS), University of California, Berkeley, January 1998. Report No. UCB/CSD-98-990.

[Mur99] K. P. Murphy. A variational approximation for Bayesian networks with discrete and continuous latent variables. In *Proceedings of Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 457–466, 1999.

[Nea93] R. M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical report, Department of Computer Science, University of Toronto, 1993. Technical Report CRG-TR-93-1.

[Pea88] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, San Mateo, CA, 1988.

[SP90] R. D. Shachter and M. A. Peot. Simulation approaches to general probabilistic inference on belief networks. In M. Henrion, R. D. Shachter, L. N. Kanal, and J. F. Lemmer, editors, *Uncertainty in Artificial Intelligence 5*, pages 221–231, Amsterdam, 1990. North-Holland.

[Wie00] W. Wiegerinck. Variational approximations between mean field theory and the junction tree algorithm. In *Proceedings of the 16th Annual Conference on Uncertainty in Artificial Intelligence (UAI-00)*, 2000.