Piilomuuttujien oppimisesta

Mikko Koivisto

15. maaliskuuta 2002

Älykkäiden järjestelmien tutkimusseminaari
Tietojenkäsittelytieteen laitos
Helsingin yliopisto

1 Johdanto

Koneoppimisen ja tilastotieteen keskeisimpiin tutkimuskohteisiin kuuluu systemaattisten menetelmien kehittäminen mittaus- tai havaintoaineistojen käsittelyyn, data-analyysiin. Tilanteesta riippuen data-analyysin tavoitteena voi olla ongelmakentän deskriptiivinen kuvaus tai (tulevaisuuden) ennustaminen. Molemmissa tapauksissa data esitetään tyypillisesti joukkona havaintovektoreita, joista kukin on jono ennalta valittujen, ongelmakenttään liittyvien, attribuuttien havaittuja arvoja.

Data-analyysin kannalta keskeinen ja hyvin syvällinen ongelma on, että ennalta valittujen attribuuttien joukko on kiinteä (käytännössä äärellinen). Jo lähtökohtaisesti tarkasteltavaan ilmiöön liittyvien kausaliteettien ei yleensä voida olettaa selittyvän valitulla joukolla attribuutteja, jolloin tavallisimmin päädytään todennäköisyysmallintamiseen (tilastolliseen mallintamiseen). Toisaalta, vaikka ilmiön kannalta keskeiset attribuutit olisivat valittuina, voi olla, että joistakin attribuuteista saadaan puutteellisia havaintoja tai ei havaintoja laisinkaan. Viimeksi mainittu tapaus on erityisen kiinnostava, koska siinä attribuutin ei tarvitse olla edes periaatteessa havainnoitava – attribuutti on enemmän tai vähemmän mallintajan keinotekoisesti mukanaan tuoma lisä datan tarkasteluun. Tavallisimmin havaitsemattomilla attribuuteilla on kiinnitetty tulkinta ja ne ovat periaatteessa havainnoitavissa. Esimerkiksi klassisessa tilastotieteessä usein estimoitavat populaatioparametrit ovat tällaisia attribuutteja. Toisinaan mallintaja haluaa kuitenkin tarkastella ns. piilomuuttujia (engl. hidden variable). Piilomuuttujilla tarkoitetaan tässä attribuutteja, jotka eivät näy datassa, joille mallintaja ei ennalta kiinnitä tulkintaa, mutta joista mallintaja silti on suoraan (ongelmakentän kuvaaminen) tai välillisesti (ennustaminen) kiinnostunut.

Tässä kirjoitelmassa pohditaan piilomuuttujien roolia Bayesiläisessä data-analyysissä. Tarkastelu rajoitetaan menetelmiin, jotka pohjautuvat Bayes-verkkojen malliperheeseen. Näinollen jäljempänä attribuuttien sijaan puhutaan (satunnais)muuttujista. Erityisesti keskitytään piilomuuttujien oppimiseen liittyviin kysymyksiin. Ensin tarkastellaan verkkorakenteen oppimisongelmaa ja -menetelmiä yleisesti. Sitten esitellään erikseen menetelmä piilomuuttujien löytämiseen [3] ja piilomuuttujien tilajoukkojen suuruuksien oppimiseen [2].

2 Bayes-verkoista

2.1 Bayes-verkot ja datan uskottavuus

Olkoon $X_{[k]} = \{X_1, \ldots, X_k\}$ joukko satunnaismuuttujia. Bayes-verkko joukolle $X_{[k]}$ on pari (G, θ), missä G on solmujoukon $X_{[k]}$ suunnattu syklitön verkko ja $\theta = (\theta_1, \ldots \theta_k)$ on parametrivektori, jossa θ_i identifioi todennäköisyysjakauman $P_{\theta_i}(X_i \mid \mathrm{Pa}_i)$. Tässä $\mathrm{Pa}_i = \{Y \in X_{[k]} : (Y, X_i) \in G\}$ on muuttujan X_i vanhempien joukko verkossa G. Bayes-verkko (G, θ) määrittelee muuttujien yhteisjakauman luonnollisella tavalla,

$$P_{(G,\theta)}(X_1, \ldots, X_k) = \prod_{i=1}^{k} P_{\theta_i}(X_i \mid \mathrm{Pa}_i).$$

Toisin sanottuna: verkkorakenne G koodaa muuttujien ehdolliset riippumattomuusoletukset ja parametrit θ määrittelevät ehdolliset todennäköisyysjakaumat.

Kun datan $D = (D_1, \ldots, D_n)$ datavektorit ajatellaan tietyssä mielessä riippumattomiksi, datan todennäköisyydeksi saadaan

$$P_{(G,\theta)}(D_1, \ldots, D_n) = \prod_{j=1}^{n} P_{(G,\theta)}(D_j),$$

missä kukin D_j on vektori muuttujien $X_1, \ldots, X_k$ arvoja. Näin Bayes-verkko määrittelee uskottavuusfunktion (engl. likelihood function) ja kaikkien Bayes-verkkojen joukko, merkitään $\mathcal{B}(X_{[k]})$, parametrisoi jakaumaperheen. Datan perusteella "paras" Bayes-verkko voidaan määritellä suurimman uskottavuuden periaatteeseen nojautuen.

2.2 Mitä bayesiläistä on Bayes-verkossa?

Bayes-verkko voidaan nähdä myös bayesiläisenä mallina. Tällöin parin (G, θ) tarkoituksena on määritellä subjektiivinen todennäköisyysjakauma $P_{(G,\theta)}(X_1, \ldots, X_k)$. On syytä huomata, että tällöin (G, θ) on kerta kaikkiaan kiinnitetty, eikä verkkoa siten voida "oppia datasta." Tällainen Bayes-verkkojen käyttö rajoittuu joihinkin asiantuntijajärjestelmiin (ks. esim. [11]). Data-analyysiin, jossa tavoitteena on muuttujien välisten riippuvuuksien hahmottaminen, se ei sovellu.

3 Bayes-verkkojen bayesiläinen oppiminen

3.1 Bayesiläinen lähestymistapa oppimiseen

Vain sellaista voidaan oppia, josta ollaan epävarmoja. Tämä on bayesiläisen oppimisen lähtökohta. Havaintojen avulla epävarmuus vähenee, mutta harvoin häviää kokonaan.

Bayes-verkkojen oppimisen suhteen tämä tarkoittaa seuraavaa. Lähtökohtana on a priori -uskomus, joka esitetään priorijakaumana $P(G, \theta)$; sekä verkkorakennetta että parametreja tarkastellaan siis satunnaismuuttujina. Datan ja Bayesin säännön avulla priori päivittyy posteriorijakaumaksi

$$P(G, \theta \mid D) = \frac{P(G, \theta)P(D \mid G, \theta)}{P(D)} \propto P(G)P(\theta \mid G)P(D \mid G, \theta).$$

On syytä korostaa, että bayesiläiselle oppiminen tarkoittaa todennäköisyysjakauman päivittämistä, ei yhden "estimaatin" valintaa. Tällainen valinta suoritetaan vain, jos on pakko, esimerkiksi ennustustehtävässä tai yleisemmin päätösongelmassa. Valintaan ei silloin vaikuta yksin todennäköisyydet vaan myös päätökseen liitetty hyötyfunktio, jota ilman ei tietenkään ole perusteltua puhua huonoista tai hyvistä valinnoista. Kun siis halutaan oppia verkkorakenne, bayesiläinen on tyytyväinen verkkorakenteen posteriorijakaumaan:

$$P(G \mid D) = \int P(G, \theta \mid D) d\theta \propto P(G) \int P(D \mid G, \theta) dP(\theta \mid G).$$

Jos kuitenkin on valittava jokin verkkorakenne, hyötyfunktiota määrittelemättä voidaan valita vaikkapa todennäköisin verkkorakenne

$$\hat{G} = \arg\max_{G} P(G \mid D),$$

missä G käy läpi kaikki verkkorakenteet. Tällöin puhutaan ns. bayesiläisen pisteytyksen (engl. Bayesian scoring metric) maksimoimisesta [8, 3, 2]. Tehtävänasettelua voidaan kritisoida: voiko käytännön kannalta olla kiinnostavaa "oppia" yksi paras verkko, jonka todennäköisyys on kuitenkin mitätön, pienempi kuin 0.0001? (Näin voinee helposti käydä, kun muuttujia on paljon mutta dataa vähän [5, 6].)

3.2 Piilomuuttujien oppiminen

Edellä kuvattu lähestymistapa toimii, vaikka opittava verkko sisältäisi piilomuuttujia. Tämä edellyttää kuitenkin, että piilomuuttujista on jo ennalta jonkinlainen käsitys. On nimittäin kiinnitettävä piilomuuttujien lukumäärä ja tilajoukot (niiden koko). Lisäksi on määrättävä priori, nyt piilomuuttujia sisältäville, verkkorakenteille ja parametreille. Jos nämä tekijät saadaan määrättyä, piilomuuttujien oppiminen palautuu Bayes-verkkojen oppimiseen Bayes-verkkojen luokassa $\mathcal{B}(X_{[k+h]})$, missä $\{X_{k+1}, \ldots, X_{k+h}\}$ on piilomuuttujien joukko.

Mielenkiintoisemmaksi tilanne menee, jos piilomuuttujien lukumäärää ja tilajoukkojen kokoa ei tunneta. Tällöin oppimista ei enää luontevasti hallita yhden kiinteän satunnaismuuttujajoukon määräämässä Bayes-verkkojen luokassa. Riittää kuitenkin tarkastella Bayes-verkkojen luokkien yhdistettä $\bigcup_{h=0}^{\infty} \mathcal{B}(X_{[k+h]})$, mikäli kunkin piilomuuttujan X_i tilajoukon koko koodataan[1] parametreihin θ_i. Tähän lähestymistapaan perustuvaa oikeaoppista Bayes-oppimista ei liene kuitenkaan enempää tutkittu eikä kokeiltu. Sen sijaan on kehitetty heuristiikoiksi käsitettäviä menetelmiä, joissa etsintämenetelmä ja menetelmän tavoite kietoutuvat valitettavalla tavalla toisiinsa [3, 2]. Näitä menetelmiä tarkastellaan lähemmin osioissa 4 ja 5.

Eräässä mielessä piilomuuttujia ei edes ole tarvetta oppia. Jos piilomuuttujia ei milloinkaan havaita, ei niiden olemassaolo, saatika oppiminen, ole oleellista, kuten Elidan et al. [3] myöntävät. Riittää yksinkertaisesti tarkastella niiden muuttujien jakaumaa, jotka ovat relevantteja kyseisen päätösongelman kannalta. Piilomuuttujien oppimista voidaan silti perustella. Yhtäältä piilomuuttujat voidaan nähdä teknisenä apuvälineenä priorijakaumien esittämisessä. Toisaalta piilomuuttujien ”löytäminen” voi auttaa ongelmakentän kuvailussa, esimerkiksi uusien käsitteiden tai suureiden keksimisessä. On myös väitetty, että piilomuuttujien löytäminen parantaa ennustamiskykyä [3, 2].

3.3 Laskentamenetelmistä

Edellä tarkasteltiin Bayes-verkkojen oppimista puhtaasti periaatteellisesta näkökulmasta. Jotta menetelmiä voitaisiin soveltaa käytännössä, on löydettävä keinoja tarvittavien suureiden eksaktiin tai likimääräiseen laskemiseen.

Tarkastellaan posteriorijakauman $P(G, \theta \mid D)$ oppimista. Ketjusäännön

[1] Näin voidaan tehdä ainakin, jos tilajoukon koko on äärellinen ja ehdollinen todennäköisyysjakauma on multinomijakauma.

nojalla yhteisjakauma voidaan koostaa kahdesta osasta,

$$P(G, \theta \mid D) = P(G \mid D)P(\theta \mid G, D).$$

Laskennallisen käsittelyn kannalta tulon kaksi tekijää poikkeavat toisistaan. Jälkimmäinen termi, parametrien posteriorijakauma annetussa verkkorakenteessa, on yleensä kohtuullisen hyvin hallittavissa joko eksaktisti tai likimääräisesti, tietyissä tapauksissa jopa suljetussa muodossa [4]. Termeistä ensimmäinen, verkkorakenteen posteriorijakauma, on vaikeammin hallittava. Syynä tähän on verkkorakenteiden diskreetti luonne ja suuri lukumäärä. Madigan ja York [9] ovat esittäneet ratkaisuksi Markovin ketju Monte Carlo -menetelmää (lyh. engl. MCMC, ks. esim. [7]). Menetelmä tuottaa stokastisella simuloinnoilla posteriorijakaumasta suurehkon otoksen verkkorakenteita. Satunnaisotosta voidaan käyttää edustamaan todellista (analyyttistä) jakaumaa, esimerkiksi bayesiläisessä ennustamisessa tai verkkorakenteen kiinnostavien piirteiden päättelyssä.

Verkkorakenteiden posteriorijakauma $P(G \mid D)$ on kiinnostuksen kohteena myös, kun maksimoidaan bayesiläistä pisteytystä. Tällöin ei kuitenkaan tarvitse hallita koko jakaumaa vaan riittää, että löydetään globaali tai hyvä lokaali maksimipiste (verkkorakenne). Friedman [4] esittää tähän päämäärään pyrkivän algoritmin, joka toimii myös piilomuuttujia sisältävässä verkossa, toisin kuin eräät aikaisemman menetelmät (esim. [8]). Tätä menetelmää tarkastellaan lähemmin osiossa 4.

Toisin kuin edellä ehkä annettiin ymmärtää, useinkaan kiinnostus ei ole Bayes-verkoissa tai verkkorakenteissa itsessään. Pikemminkin tavoitteena on oppia joitakin ominaisuuksia, jotka ovat yhteisiä todennäköisille verkoille. Ennustustehtävässä niinikään halutaan lopulta laskea todennäköisyyksia (tulevalle) datalle, ei verkoille. Näissä tapauksissa Bayes-verkkojen tai verkkorakenteiden posteriorijakauman eksplisiittinen hallitseminen voidaan sivuuttaa, sillä lopulta Bayes-verkot ”integroidaan pois”. Seuraavassa tarkastellaan tätä mahdollisuutta lähemmin.

Olkoon kiinnostuksen kohteena verkkorakenteiden funktio f. Tällöin laskettava suure on f:n posterioritodennäköisyys,

$$P(f \mid D) = \sum_G P(G \mid D) P(f \mid G).$$

Tässä G käy läpi kaikki verkkorakenteet. Yllä mainittu Madiganin ja Yorkin [9] MCMC-menetelmä tarjoaa suoraviivaisen keinon summan laskemiseen: Monte Carlo -integrointi nojautuen otokseen verkkorakenteita. Hyvän otoksen tuottaminen ei kuitenkaan ole ongelmatonta. Parannukseksi tähän Friedman ja Koller [5] ehdottavat menetelmää, joka perustuu muotoiluun

$$P(f \mid D) = \sum_{\prec} P(\prec \mid D) P(f \mid \prec, D),$$

missä $\prec$ käy läpi kaikki tarkasteltavien muuttujien lineaarijärjestykset (verkkorakenteen suunnatut kaaret). Osoittautuu, että sopivilla f todennäköisyydet $P(D \mid \prec)$ ja $P(f \mid \prec, D)$ voidaan laskea tehokkaasti, toisinaan jopa suljetussa muodossa. Taustalla on oivallus, että termeihin sisältyvä summaus yli verkkorakenteiden voidaan hajottaa riippumattomiksi summauksiksi yli muuttujakohtaisten "perheiden", mikäli muuttujien järjestys on kiinnitetty [1, 5]. Nyt likimääräinen laskenta voidaan suorittaa Monte Carlo -integroinnilla nojautuen otokseen lineaarijärjestyksiä. Vaikka menetelmä on toteutettu ainoastaan täydellisille havaintoaineistoille, Friedman ja Koller arvelevat sen soveltuvan pienillä muutoksilla myös puuttuvan datan ja piilomuuttujien tapauksessa [5].

4 Menetelmä riippuvuusrakenteen oppimiseen

Elidan et al. [3] tarkastelevat piilomuuttujien oppimista tilanteessa, jossa muuttujien lukumäärä ja sijainti (so. suhteet muihin muuttujiin) ovat tuntemattomia. Piilomuuttujien tilajoukot lienevät heillä kiinnitetyt binäärisiksi; tämä seikka ei kuitenkaan selviä kunnolla heidän esityksestään.

Kirjoittajien ehdottama menetelmä voidaan nähdä verkkorakenteen oppimiseen tarkoitetun rakenne-EM -algoritmin (engl. structural EM) laajennuksena [4]. Annettuna data rakenne-EM -algoritmi yrittää löytää Bayes-verkkorakenteen, joka maksimoi bayesiläisen pisteytyksen. Aloittaen annetusta aloitusverkkorakenteesta, algoritmi suorittaa iteratiivisesti seuraavia askelia:

E-askel: Laske alarajafunktio kaikille verkkorakenteille G (käytännössä vain joillekin ”naapuriverkoille”):

$$Q(G : G_t) = E\{\log P(G \mid D)\},$$

missä odotusarvo otetaan yli puuttuvien data-alkioiden jakauman ehdolla edellisen iteraation tuottama verkkorakenne G_t.

M-askel: Aseta

$$G_{t+1} = \arg\max_G Q(G : G_t).$$

Kun verkko ei enää parane (oleellisesti), algoritmi lopettaa ja palauttaa viimeisimmäksi löydetyn verkon.

Elidan et al. toteavat, että edellä kuvattu algoritmi onnistuu piilomuuttujien sijoittelussa vain, jos piilomuuttujien on ennalta, aloitusverkkorakenteessa, osattu sijoittaa jotakuinkin kohdalleen [3]. Tämän harmillisen vaatimuksen poistamiseksi he ehdottavat E-askeleen muokkaamista. Ajatuksena on löytää havaittujen muuttujien verkkorakenteesta rakenteellisia merkkejä (engl. structural signature), jotka kertovat piilomuuttujien mahdollisista suhteista havaittuihin muuttujiin. Tällainen merkki on heidän mielestään verkosta löytyvä puoliklikki (engl. semi-clique), jossa jokaisesta solmusta on yhteys vähintään puoleen klikin muista solmuista. Intuitio on, että klikkiä vastaava vahva riippuvuusrakenne ei ole hyvä selitys – parempi on esitellä piilomuuttuja purkamaan ja yksinkertaistamaan kyseinen monitahoinen riippuvuusrakenne. Kun piilomuuttuja on kerran lisätty tarkasteltavaan muuttujajoukkoon, sitä ei enää oteta pois. Uusia piilomuuttujia voidaan kylläkin oppia EM-algoritmin edetessä.

Elidanin et al. vertailussa [3] puoliklikkien etsiminen näytti toimivan hyvin. Menetelmä pärjäsi hyvin niin bayesiläisen pisteytyksen maksimoinnissa kuin myös ennustuksessa (log-loss -kustannusfunktiolla).

5 Menetelmä tilajoukkojen suuruuksien oppimiseen

Edellä kuvattu menetelmä verkkorakenteen oppimiseen ja piilomuuttujien löytämiseen oletti, että piilomuuttujien tilajoukon koko on kiinnitetty. Käytännössä tilajoukkoa tai sen kokoa ei kuitenkaan useinkaan tunneta. Onkin tarpeellista pohtia, voidaanko tilajoukon koko jotenkin oppia datasta.

Elidan ja Friedman [2] ehdottavat tilajoukkojen koon oppimiseen lähestymistapaa, joka perustuu edellisessä osiossa kuvatulle piilomuuttujien paikannusmenetelmälle. Heidän perusajatuksensa on seuraava. Tarkastellaan tilannetta, jossa verkkorakenne piilomuuttujineen on kiinnitetty joillakin sopivilla tilajoukkojen suuruuksilla. Tilajoukkojen suuruuksien ”optimoimiseksi” suoritetaan seuraava algoritmi:

1. Aseta kaikille piilomuuttujille yksi tila (siis tilajoukon koko on yksi). Generoi jokaiselle piilomuuttujalle kuvitteelliset havainnot (yksi ja sama havainto jokaisessa todella havaitussa datavektorissa). Merkitse kaikki piilomuuttujat selvittämättömiksi.

2. Valitse (satunnaisesti tai järjestyksessä) yksi selvittämätön piilomuuttuja tarkasteluun.

3. Aseta piilomuuttujalle maksimaalinen määrä tiloja: yksi per muuttujan Markov-peitteen jäsenten datassa esiintyvä arvoasetus. Generoi vastaavat kuvitteelliset havainnot tarkasteltavalle piilomuuttujalle.

4. Etsi piilomuuttujan kaksi tilaa, joiden yhdistäminen tuottaa suurimman bayesiläisen pisteytyksen. Yhdistä löydetyt tilat. Laske piilomuuttujalle kuvitteellinen havainto sulauttamalla yhdistettyjä tiloja vastaavat havainnot. Talleta pisteytys.

5. Jatka kohdasta 4, kunnes tiloja on vain yksi.

6. Aseta piilomuuttujan tilajoukoksi paras löydetty.

7. Jos piilomuuttujan tilajoukko muuttui aikaisempaan verrattuna, merkitse jokainen valitun piilomuuttujan Markov-peitteeseen kuuluva piilomuuttuja selvittämättömäksi.

8. Jatka kohdasta 2, kunnes jokainen piilomuuttuja on selvitetty.

9. Palauta löydetyt tilajoukot.

Algoritmi löytää hyvät tilajoukot piilomuuttujille syötteenä annetussa verkkorakenteessa. Rakenne-EM -algoritmi, paitsi tuottaa aloitusverkon, voi myös jatkaa saadusta verkosta ja löytää parempia rakenteita. Tällöin tilajoukot joko pidetään kiinnitettyinä tai edellä kuvattua algoritmia sovelletaan toistuvasti EM-algoritmin osana. Koska piilomuuttujille kiinnitettiin tässä kuvitteelliset havainnot, EM-algoritmin E-askeleesta jää pois odotusarvon ottaminen yli näiden tuntemattomien arvojen.

Elidanin ja Friedmanin mukaan yllä kuvattu ahne algoritmi on tehokas vaihtoehto suoraviivaisemmalle EM-algoritmin laajennukselle, jossa rakenne-EM -algoritmi suoritetaan erikseen kullekin mahdolliselle tilajoukkojen asetukselle, ja lopuksi valitaan paras yhdistelmä. Kirjoittajien testeissä ero tulosten hyvyydessä on pieni mutta ero laskenta-ajassa suuri.

6 Yhteenveto

Tässä kirjoitelmassa tarkasteltiin Bayes-verkkojen oppimista. Erityisesti keskityttiin bayesiläisen oppimisen periaatteisiin ja piilomuuttujien rooliin. Laskennallisesti raskaiden lähestymistapojen vaihtoehtona esiteltiin Elidanin et al. menetelmät piilomuuttujien löytämiseen ja paikantamiseen verkkorakenteessa [3] sekä piilomuuttujien tilajoukkojen koon optimointiin [2]. Menetelmät on nähtävä jatkona Fridmanin [4] rakenne-EM -algoritmille.

Elidanin et al. menetelmillä on kaksi heikkoutta. Ensinnäkin ne ovat ahneita heuristiikkoja, joiden suorituskyky ei ole mitenkään taattu. Kokeet näyttäisivät osoittavan, että menetelmä toimii hyvin, mutta tuloksia latistaa se, että kirjoittajat eivät pohdi piilomuuttujille asetettavien priorien vaikutusta. Toiseksi algoritmit pyrkivät maksimoimaan bayesliläistä pisteytystä, mikä ei useinkaan liene hyvin perusteltu tavoite.

Kiinnostava vaihtoehto Elidanin et al. menetelmille on lähestymistapa, jossa bayesiläisen pisteytyksen maksimoinnin sijaan pyritään hahmottamaan havaittavien muuttujien sekä piilomuuttujien lukumäärän, sijoittelun ja tilajoukkojen koon yhteisposteriorijakauma. Vaihtoehtoisesti voitaisiin keskittyä joihinkin jakauman kiinnostaviin marginaaleihin. Tämä lähestymistapa on saanut yllättävän vähän huomiota, vaikka Madigan ja Yorkin [9] tarjoamalla MCMC-menetelmällä on kohta ikää jo kymmenen vuotta. Lähestymistapa on kuitenkin lähes väistämätön esimerkiksi geeniekspressiodatan analyysissä, jossa havaintoaineisto on tyypillisesti todella harva [5, 6, 10]. Piilomuuttujien oikeaoppinen käsittely vaatinee kuitenkin menetelmien lisäkehittelyä [10].

Ensituntumalta piilomuuttujien tarkastelu vaikuttaisi bayesiläisessä lähestymistavassa tarpeettomalta, mikäli mahdolliset piilomuuttujat eivät ilmene lopullisessa päätösongelmassa. Tarkemmin ajateltuna piilomuuttujilla on kuitenkin mielenkiintoinen, jopa tärkeä rooli. Bayesiläisessä päättelyssä on nimittäin aina rakennettava priorijakauma, ja tässä tehtävässä piilomuuttujat ovat paikallaan. Miksi piilomuuttujien löytäminen ja käyttäminen voi parantaa ennustuksia, paitsi heuristissa menetelmissä kuten [2], myös oikeaoppisessa bayesiläisessä ennustamisessa? Siksi, että alunperin huolimattomasti valittu priorijakauma saa tällöin järkevämmän muodon!

Viitteet

[1] Buntine, W., Theory refinement of Bayesian networks. *Proc. of 7'th Annual Conf. on Uncertainty in Artificial Intelligence (UAI-91)*, 1991.

[2] Elidan, G., Friedman, N., Learning the dimensionality of hidden variables. *Proc. of 17'th Annual Conf. on Uncertainty in Artificial Intelligence (UAI-01)*, 2001.

[3] Elidan, G., Lotner, N., Friedman, N., Koller, D., Discovering hidden variables: a structure-based approach. *NIPS*, 2000.

[4] Friedman, N., The Bayesian structural EM algorithm. *Proc. of 16'th Annual Conf. on Uncertainty in Artificial Intelligence (UAI-98)* 1998.

[5] Friedman, N., Koller, D., Being Bayesian about network structure. *Proc. of 16'th Annual Conf. on Uncertainty in Artificial Intelligence (UAI-00)*, 2000.

[6] Friedman, N., Linial, M., Nachman, I., Pe'er, D., Bayesian networks to analyze expression data. *RECOMB*, 2000.

[7] Gilks, W. R., Richardson, S., Spiegelhalter, D. J., *Markov Chain Monte Carlo in Practice.* Chapman & Hall, Lontoo, Iso-Britannia, 1996.

[8] Heckerman, D., Geiger, D., Chickering, D. M., Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20 (1995), 197-243.

[9] Madigan, D., York, J., Bayesian graphical models for discrete data. *Inter. Stat, Rev.* 63 (1995), 215–232.

[10] Murphy, K. P., Learning Bayes nets structure from sparse data sets. Tekninen raportti, 2001.
Saatavilla osoitteessa www.cs.berkeley.edu/∼murphyk/papers.html

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