

1. fejezet

Egzakt, optimalizációs, és Monte Carlo következtetések VGM-benn

1.1. Prediktív következtetés Bayes-hálókbán

Mint láttuk, a Bayes-háló elsődleges célja, hogy az általuk reprezentált valószínűségi változók együttes eloszlását leírja. A Bayes-hálókbán alkalmazott legalapvetőbb művelet a *prediktív következtetés*, amely során egy vagy több ún. *célváltozó* eloszlását vagy adott konfigurációjuk valószínűségét keressük adott feltételek mellett. Ezek a feltételek tipikusan az ún. *bizonyíték-* vagy *evidencia* változókra vonatkozó ismeretek.

Ebben a fejezetben a diszkrét változókat tartalmazó Bayes-hálókbán való következtetéssel foglalkozunk.

A bizonyíték típusai Ha egyértelműen ismert a bizonyítékváltozó által felvett érték, akkor *biztos evidenciáról*, ha csak egy eloszlás erejéig ismert, akkor *bizonytalan evidenciáról* beszélünk. Látható, hogy a biztos evidencia a bizonytalan speciális esete: ekkor a változó eloszlása $0, \dots, 0, 1, 0, \dots, 0$ alakú.

A két legáltalánosabb következtetési eset az alábbi:

Egy célváltozó marginálisa. A következtetés alapfeladata: keresett az egyetlen célváltozó eloszlása a bizonyíték-ismeretek, mint feltétel mellett ($\mathbf{P}(X|\mathbf{E} = \mathbf{e})$). A fő oka annak, hogy ezt a következtetési esetet külön kiemeljük az, hogy sok következtetési algoritmus (elsősorban az egzaktak) esetén a legegyszerűbb és leghatékonyabb módja az összetettebb esetek kezelésének az, ha azt visszavezetjük erre.

Több célváltozó együttes konfigurációja. A következtetés általános esete: az evidenciák ismeretében keresett a (tipikusan) több célváltozó együttes eloszlása. Ha egy adott algoritmus esetén nem adható közvetlenül hatékony megoldás erre az esetre, akkor alkalmazható a láncszabály: vesszük az első célváltozót és kiszámítjuk a valószínűségét a bizonyítékok mellett; ezután a bizonyítékok halmazát kibővítjük a célváltozó értékével és a következő célváltozó valószínűségét már az új bizonyíték

mellett számítjuk. Ha végigértünk a teljes célváltozó-halmazon, a keresett valószínűséget az egyes elemekre kapott részeredmények szorzataként kapjuk.

A fenti alapesetek alkalmazásaként még számos következtetési típust kaphatunk. Ezek közül a legjelentősebbek a következők:

Következtetés érzékenysége (Sensitivity of inference) Ebben az esetben azt vizsgáljuk, hogy adott változók értékére vonatkozó ismereteink, hogyan befolyásolják a célváltozók eloszlását. Azaz egy vizsgált változó esetén azt vetjük össze, hogy a célváltozó(k)nak az evidenciák eredeti állapota szerinti feltételes eloszlása hogyan változik, ha a bizonyítékok körét rendre kibővítjük a vizsgált változó értékeivel. A vizsgálat természetesen kibővíthető több vizsgált változóra is.

Többletinformáció értéke (Value of information) Amennyiben a háló csomópontjainak (egy halmazának) konfigurációihoz valamiféle hasznosságértéket rendelünk, vizsgálhatjuk, hogy egy adott (vagy több) változó értékének ismerete hogyan fogja megváltoztatni a várható hasznosságot. Az információ értéke a következő képlettel számolható:

$$VOI(X|E) = \sum_{x \in X} P(x|E) \times |E_{\{V \setminus E \setminus X\}}[U(V)] - E_{\{V \setminus E\}}[U(V)]|; \quad (1.1)$$

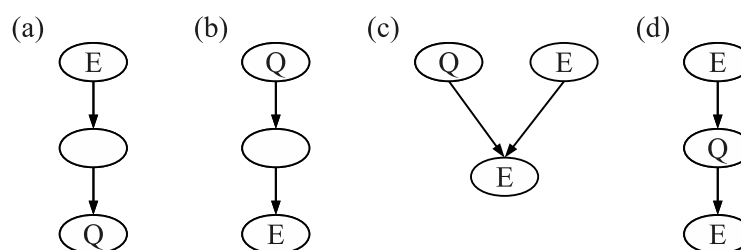
vagyis várhatóan (az új evidencia feltételes valószínűségével súlyozva) mekkora abszolút hasznosságváltozást fog okozni az új bizonyíték belépése.

1.2. A következtetési eljárások áttekintése

Bár a Bayes-hálókból való prediktív következtetés célja minden esetben alapvetően az, hogy bizonyos ismeretek birtokában a háló többi részére vonatkozó valószínűségi mennyiségeket számítsunk ki, magára a következtetés elvégzésére számos eljárás alkalmazható, aszerint, hogy mi a következtetés konkrét tárgya. Ennek megfelelően a következtetési eljárásokat többféle szempont szerint is csoportosíthatjuk, melyek közül a leglényegesebbek a következők:

- Az oksági viszonyok, azaz a különböző szerepű (megfigyelés illetve célpont) csomópontoknak a hálóban egymáshoz képest elfoglalt strukturális helyzete alapján;
- Az alkalmazott eljárás jellege szerint;
- Illetve a Bayes-háló (és ezzel együtt a teljes következtetés) komplexitása szerint.

A megfigyelés - és a célsomópontok oksági viszonya a intuitíven fogalmazva a következtetés logikai irányát jelenti, azaz hogy a cél- és a bizonyítékváltozók között menő utakon milyen az élek irányítottsága. E szerint a besorolás szerint a fő csoportok a következők (az 1.1 ábra illusztrálja ezeket):



1.1. ábra. Tipikus következtetési esetek az evidenciák és célváltozók topologikus viszonya szerint: okozati (a), diagnosztikai (b), okok közti kimagyarázás (c) és kevert (d).

Okozati A megfigyelt csomópont a célpontnak az őse, azaz a következtetés iránya megegyezik a háló definiálása során feltett ok-okozati iránynak: az okot ismerve kell meghatároznunk az okozat eloszlását.

Diagnosztikai Az előző ellentettje, bizonyos következmények ismeretében kell meghatároznunk az azt kiváltó okokat (pontosabban azok eloszlását).

Okok közötti kimagyarázás A háló egy csomópontjának két őse között (egy v-struktúra két felső csomópontja) között is fellép valószínűségi függés, amennyiben a közös lezármazott értéke is ismert. Ilyen esetben beszélhetünk egy ok kimagyarázásáról, azaz, amikor egy másik ok ismerete a vizsgált ok eloszlását megváltoztatja.

Kevert Általános esetben (amikor több megfigyelési és célpont változónk is van) a megfigyelések és célpontok strukturális viszonya nem fedhető le teljes mértékben egyik fenti kategóriával sem, ilyenkor beszélünk *kevert* következtetésről.

1.2.1. A következtetési algoritmus

Az alábbiakban a legalapvetőbb, illetve a leggyakrabban alkalmazott következtetési eljárásokat soroljuk fel.

Egy teljes konfiguráció valószínűségének kiszámítása. Amennyiben a megfigyelt - és a célváltozók halmazai együtt lefedik a háló egészét, a kérdéses valószínűség a Bayes-hálók definíciójául is szolgáló szorzatképlet egyszerű módosításával számítható:

$$P(\mathbf{V} = \mathbf{v}) = \prod_{X \in \mathbf{V}} P(X = x | Pa(X)); \quad (1.2)$$

azaz a változók egy topologikus sorrendjén végighaladva, rendre értékül adjuk az adott változónak a megfigyelt vagy lekérdezendő értéket, és amennyiben a csomópont célváltozó, úgy annak feltételes valószínűségét bevesszük a szorzatba, amennyiben megfigyelt, kihagyjuk a szorzatból.

Bár ilyen következtetési eset praktikus esetben szinte sohasem fordul elő, az eljárás megjelenik a következő következtetési módszer építőkövéjeként.

Kimerítő felsorolás A legegyszerűbb következtetési eljárás: lényegében felsoroljuk a vizsgált bizonyíték-célváltozó konfigurációval kompatibilis teljes konfigurációkat és ezek valószínűségeinek összegeként adódik a keresett valószínűség. A részletes leírást az 1.3.1 alfejezet tartalmazza.

Következtetés fa-gráfokban Ha a vizsgált Bayes-háló struktúrája egy fa-gráf (vagyis bármely két csomópontja között maximum 1 út vezet) akkor lehetőség van a fenténél jelentősen hatékonyabb egzakt következtetésre. Ennek az algoritmusnak a leírását az 1.3.3 alfejezet tartalmazza.

Következtetés másodlagos struktúrában Ha a hálóban van irányítatlan kör, akkor a fa-gráfokra alkalmazható algoritmus már nem használható. A másodlagos struktúrákat létrehozó algoritmusok ezt próbálják kiküszöbölni: a hálót úgy alakítják át, hogy az új modell az eredeti együttes eloszlást reprezentálja, de egy fa-struktúrában, amelyben már hatékonyan következtethetünk. Természetesen ezeknél az algoritmusoknál a látszólagos komplexitás-csökkenésnek mindig megvan az ára: pl. az új struktúra csomópontjai az eredetihez képest jóval nagyobb értékűk lesznek.

Az ebbe a kategóriába tartozó algoritmusok közül bemutatunk néhány egyszerűbbet (1.3.4 alfejezet), illetve egy összetettebbet, amely valós alkalmazásokban a leggyakrabban elő szokott fordulni (1.4 alfejezet).

Közelítő következtetés Monte Carlo eljárásokkal Ha a fenti egzakt eljárások nem alkalmazhatók (ennek oka a leggyakrabban a háló túl nagy volta), sztochasztikus szimuláció alkalmazható. Ezen eljárások alapötlete, hogy mintavételezzük a háló által reprezentált eloszlást, és a minta alapján számított statisztikával becsüljük a keresett mennyiséget (1.5.2 és 1.5.3 alfejezet).

Természetesen itt is igaz, hogy az egyszerűbb algoritmusok költsége a probléma növekedtével kezelhetetlenül nagyra nő. Az 1.6.4 fejezetben bemutatunk egy olyan Markov-lánc Monte Carlo mintavételezési eljárást, amely nem közvetlenül a Bayes-háló együttes eloszlását mintavételezi (de belátható róla, hogy az egyensúlyi eloszlása ehhez tart), és így kezelhetőbb komplexitású megoldást kínál.

1.2.2. A következtetés komplexitása

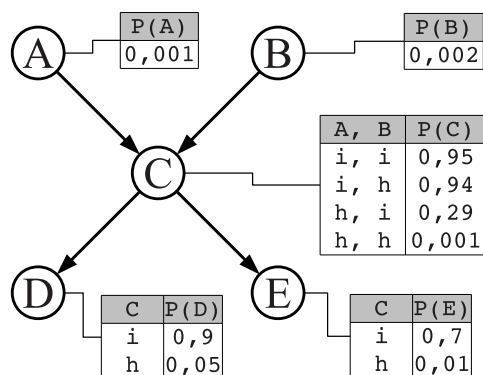
Általános esetben (amikor a Bayes-háló tetszőlegesen nagy fokban összekötött részeket is tartalmazhat) belátható, hogy a következtetés feladata NP-nehéz, és a következtetés a csomópontok számában exponenciális komplexitású. Az ezzel kapcsolatos gondolatmeneteket és levezetéseket az 1.7 függelék tartalmazza.

Következtetés polifákban. Fa-gráfokban (azaz amelyeknek bármely két csomópontja között maximum 1 út fut) belátható, hogy a következtetés végrehajtható polinom időben.

Ez igaz mind az 1.3.3 alfejezetben bemutatott speciálisan fa-gráfokra alkalmazható algoritmusra (hisz az eljárás minden egyes csomópontot maximum egyszer érint), mind az 1.3.2 alfejezetben bemutatott változó eliminációs algoritmusra. A részletes gondolatmeneteket a hivatkozott alfejezetekben közöljük.

1.3. Egyszerűbb egzakt következtető eljárások

A következőkben a Bayes-hálókból való egzakt következtetésre szolgáló alapvetőbb eljárásokat mutatunk be. Az algoritmusok többségét az alábbi modell felhasználásával illusztráljuk:



1.2. ábra

A következő algoritmusok leírásánál minden esetben az egy változó marginális eloszlását kiszámító eljárást adjuk meg; ha több változó egy együttes konfigurációjának valószínűségét akarjuk kiszámítani, az megtehető a *lányszabály* alkalmazásával: először kiszámítjuk az első célváltozó valószínűségét, ezután az evidenciák halmazát kibővítjük ezzel az értékkel és kiszámítjuk a második célváltozó valószínűségét. Ha ezzel az eljárással végigérünk az összes célváltozón, a menet közben kapott egyes valószínűségek szorzataként adódik a teljes konfiguráció valószínűsége.

$$P(X_1, \dots, X_n | E) = \prod_{i=1}^n P(X_i | E \cup \{X_j\}_{j=1}^{i-1}) \quad (1.3)$$

1.3.1. Következtetés felsorolással

A legegyszerűbb következtetési eljárásban lényegében felsoroljuk a szabad (nem cél- és nem evidencia) változók összes lehetséges konfigurációját, és azokat a célváltozó értékei szerint szeparálva összegezzük. Az egyes célváltozó-értékekhez tartozó összegek a célváltozó nor-

malizálatlan eloszlását adják, amiből a normalizált értékek már egyszerűen számíthatók:

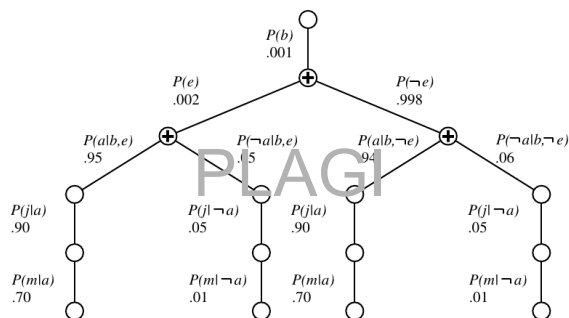
$$P(X = x|E = e) = \sum_{F_1} \cdots \sum_{F_n} \prod_{i=1}^N \underbrace{P(V_i|\{V_j\}_{j=1}^{i-1})}_{P(V_i|Pa(V_i))}, \quad (1.4)$$

ahol $\{F_i\}$ a „szabad”, azaz se nem cél- se nem evidencia változók halmaza.

Valójában nem kell, hogy a szorlatban szereplő összes tag az összes szummán belül helyezkedjen el: ha a változók felett futó szummákat a Bayes-háló egy topologikus sorrendjében (\prec_{top}) helyezzük el, a $P(V_i|Pa(V_i))$ tagok előrehozhatók közvetlenül a vonatkozó \sum_{V_i} szumma mögé, hisz az értékük nem függ a szumma indexváltozójától, így konstansként kiemelhetők. Így a kifejezés a következő lesz:

$$P(X = x|E = e) = \prod_{i:V_i \prec_{top} F_1} P(V_i|Pa(V_i)) \times \sum_{F_1} \left(\prod_{i:V_i \prec_{top} F_2} P(V_i|Pa(V_i)) \times \sum_{F_2} (\dots) \right) \quad (1.5)$$

Magát a kifejezést és annak kiértékelését az 1.3 ábra szemlélteti, kiszámítása pedig az 1.1 algoritmus szerint történhet.



1.3. ábra. A felsorolós következtetés által kiértékelt kifejezés szerkezete a $P(A|D, E)$ lekérdezés esetén.

Tehát a „FELSOROLÁS-KÉRDEZÉS” csak annyit tesz, hogy rendre kibővíti az evidenciákat a célváltozó értékeivel, ezen inicializáló lépések után pedig „FELSOROL-MINDET” számolja ki az 1.5 egyenletben szereplő összeget, azaz a normalizálatlan eloszlás-vektor elemeit.

1.3.2. Következtetés változó eliminációval

Az előbbieken bemutatott felsorolós eljárás fő gyengesége hatékonysági szempontból, hogy a kifejezés-fa alsó csomópontjaiban található mennyiségeket feleslegesen, többször is kiértékeli; természetesen adódik tehát a hatékonyságot javító módosítás: a felsorolós módszerben többször is kiértékelt mennyiségeket tároljuk el, és használjuk fel őket újra, amikor szükségesek lesznek.

Algorithm 1.1 Egzakt következtetés Bayes-hálóknban felsorolással

```

1: function FELSOROLÁS-KÉRDEZÉS( $X$ ,  $e$ ,  $bn$ )                                ▷ returns  $P(X)$ 
                                       ▷  $X$  : a lekérdezés változója
                                       ▷  $e$  : az  $\mathbf{E}$  változók megfigyelt értékei
                                       ▷  $bn$  : a valószínűségi háló

2:    $Q(X) \leftarrow$  üres eloszlás
3:   for each  $x_i$  in  $X$  értékei do
4:     terjeszd ki  $e$ -t  $x_i$ -vel
5:      $Q(x_i) \leftarrow$  FELSOROL-MINDET(VÁLTOZÓ[ $bn$ ]  $e$ )
6:   end for each
7:   return NORMALIZÁL( $Q(X)$ )
8: end function

```

```

9: function FELSOROL-MINDET( $változók$ ,  $e$ )                                ▷ returns valós szám
10:  if ÜRES( $változók$ ) then
11:    return 1.0
12:  end if
13:   $Y \leftarrow$  ELSŐ( $változók$ )
14:  if  $Y$  értéke  $y \in e$  then
15:    return  $P(y|Pa(Y)) \times$  FELSOROL-MINDET(MARADÉK( $változók$ ),  $e$ )
16:  else
17:    return  $\sum_y P(y|Pa(Y)) \times$  FELSOROL-MINDET(MARADÉK( $változók$ ),  $e \cup \{Y = y\}$ )
18:  end if
19: end function

```

A változó eliminációs eljárás által elvégzett feladat tehát a következő: adott egy Bayes-háló és ezzel együtt az általa ábrázolt $P(\mathcal{U})$ eloszlás; keressük a $P(X|E)$ eloszlást, ahol X a célváltozók, E az evidenciák halmaza. Az eljárás a számítást úgy hajtja végre, hogy \mathcal{U} -ból egymás után eliminálja a sem X -ben, sem E -ben szereplő változókat.

Példaképpen tekintsük a $P(B|j, m)$ mennyiség kiszámítását a már ismert, Judea Pearl-féle földrengéses hálóban. Számítsuk ki az alábbi kifejezést:

$$P(B|j, m) \propto \underbrace{P(B)}_B \sum_e \underbrace{P(e)}_E \sum_a \underbrace{P(a|B, E)}_A \underbrace{P(j|a)}_J \underbrace{P(m|a)}_M. \quad (1.6)$$

A kifejezés minden tag-valószínűségét megjelöltük az ahhoz kapcsolódó változó nevével, ezek az ún. *tényezők* (*factors*). A kiértékelés jobbról balra haladva, a következőképpen történik:

M: Mivel M változó értéke rögzített (a bizonyítékban) ezért itt nincs szükség M szerinti összegzésre, pusztán eltároljuk M valószínűségét, mint a feltételek (ebben az esetben A) függvényét, egy vektorban: $f'_M(A) = (P(m|a), P(m|\bar{a}))$.

J: Itt M -hez hasonlóan egy kételemű vektort kell eltárolnunk: $f'_J(A) = (P(j|a), P(j|\bar{a}))$.

A: Ebben az esetben a $P(a|B, e)$ értékeket egy $2 \times 2 \times 2$ -es „mátrixban”, $f_A(A, B, E)$ -ben tároljuk el.

\sum_a : Az A feletti összegzést két lépésben végezhetjük el: először összeszorozzuk a tényezőket, majd a szorzás eredményéből kiösszegezzük A -t. A tényezők szorzására az ún. *pontonkénti szorzás* művelet szolgál (részleteit lásd lentebb), az összegzés után pedig az $f_{AJM}(B, E)$ tényezőt kapjuk, amely tehát J és M valószínűségeit tartalmazza, B és E szerint indexelve, A -t pedig kiösszegezve (erre utal a \mathcal{A} alsó index).

... A további lépések már a fentiekből sejthetők: letároljuk f_E -t, majd összeszorozzuk az előbb kapott tényezővel és E -t kiösszegezve kapjuk $f_{E\mathcal{A}JM} = \sum_e f_E \times f_{AJM}(B, E)$ -t. Ezután már csak a B -hez tartozó tényezővel kell megszorozni az eddig kapott eredményt: $P(B|j, m) \propto f_{BE\mathcal{A}JM} = f_B \times f_{E\mathcal{A}JM}$

Pontonkénti szorzás. A fenti számításokban használt tényezők (faktorok) valójában egy változóhalmaz konfigurációival indexelt táblázatok/tömbök. A *pontonkénti szorzás* két ilyen táblázatot kombinál össze: az indexáláshoz használt változók halmaza a szorzatban a két eredetinek az uniója lesz, az egyes bejegyzések pedig az új indexből vetítéssel visszakapható eredeti indexekhez tartozó bejegyzések szorzata lesz, azaz pl. $f_{1 \times 2}(a, b, c) = f_1(a, b) \times f_2(b, c)$. Az 1.1 táblázat két tényező pontonkénti szorzatára ad egy példát.

| A | B | $f_1(A, B)$ | B | C | $f_2(B, C)$ | A | B | C | $f_{1 \times 2}(A, B, C)$ |
|-----|-----|-------------|-----|-----|-------------|-----|-----|-----|---------------------------|
| I | I | 0,3 | I | I | 0,2 | I | I | I | $0,3 \times 0,2$ |
| I | H | 0,7 | I | H | 0,8 | I | I | H | $0,3 \times 0,8$ |
| H | I | 0,9 | H | I | 0,6 | I | H | I | $0,7 \times 0,6$ |
| H | H | 0,1 | H | H | 0,4 | I | H | H | $0,7 \times 0,4$ |
| | | | | | | H | I | I | $0,9 \times 0,2$ |
| | | | | | | H | I | H | $0,9 \times 0,8$ |
| | | | | | | H | H | I | $0,1 \times 0,6$ |
| | | | | | | H | H | H | $0,1 \times 0,4$ |

1.1. táblázat. Két tényező pontonkénti szorzata.

Az eljárás komplexitása polifákban.

Látható, hogy a változó eliminációs algoritmus során egy-egy összegzés elvégzésével mindig eltávolítunk egy csomópontot a képletből, vagyis az ilyen kiösszegzések számát a változók száma felülről korlátozza. Ezek után belátható, hogy a teljes eljárás költségét az határozza meg, hogy egy-egy kiösszegzés során mekkora $f_{\dots}(\dots)$ tényezőt kell előállítani és feldolgozni. Ezek mérete erősen függ az eliminálás sorrendjétől, de ennek helyes megválasztása esetén, azaz ha ez konzisztens a háló egy topologikus sorrendezésével, a legnagyobb ilyen tényező is arányos lesz az éppen eliminált csomópont CPT-jének méretével.

Ha tehát a hálóban a csomópontonkénti szülők számára, és a csomópontok értékészletére is adott egy felső korlát (k és d), akkor az eljárás komplexitása $\mathcal{O}(nd^k)$ lesz (ahol n a csomópontok száma).

Változók relevanciája a következtetés szempontjából

Tekintsük a következő lekérdezést ugyanebben a hálóban:

$$P(J|b) \propto P(b) \sum_e P(e) \sum_a P(a|b, e) P(J|a) \sum_m P(m|a). \quad (1.7)$$

Látható, hogy a $\sum_m P(m|a)$ mennyiség definíció szerint 1 lesz, vagyis belátható, hogy minden levélcsomópont, amely nem bizonyíték, vagy célváltozó eltávolítható a hálóból a konkrét következtetési esetben, mivel nem befolyásolja annak kimenetelét. Természetesen, ha egy levélcsomópontot ily módon eltávolítottunk, az új háló is tovább „nyeshető”, amiből a következő tétel adódik.

1.1. tétel (Változók irrelevanciája). *Egy Bayes-hálóban minden csomópont, amely nem eleme vagy őse a bizonyíték- vagy a célváltozóknak, irreleváns az adott lekérdezés szempontjából.*

A releváns csomópontok halmaza tovább szűkíthető a következők figyelembevételével:

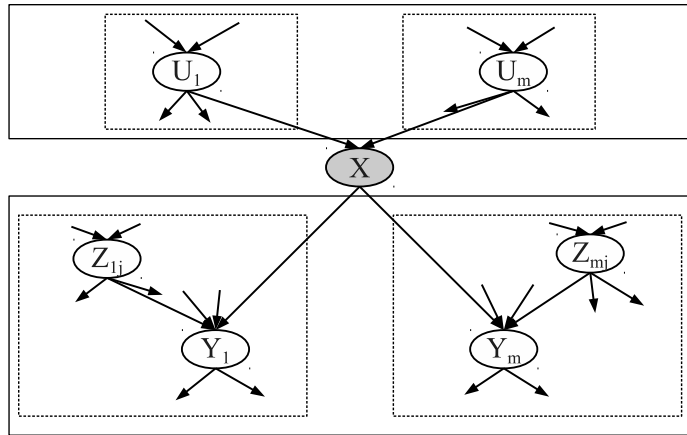
1.1. definíció (Morális gráf). Egy DAG (irányított körmentes gráf) morális gráfja úgy kapható meg, hogy az eredeti gráfban az egyes csomópontok szüleit egymással összekötjük, majd az így kapott gráfban töröljük az élek irányítását.

1.2. definíció (m-szeparáció). Az X és Y változóhalmazokat Z m-szeparálja a G DAG-ban, ha Z szeparálja X -et és Y -t G morális gráfjában.

1.2. tétel (Változók irrelevanciája II.). Azon változók, amelyeket G -ben a bizonyítékok m-szeparálnak a célváltozóktól, az adott lekérdezés szempontjából irrelevánsak.

1.3.3. Következtetés polifákban

Ha a Bayes-háló egy ún. erdő, azaz egy (potenciálisan) több fából álló gráf-, akkor létezik a változók számában lineáris következtetési algoritmus. Ahhoz hogy ezt belássuk, tételezzük fel, hogy a feladat egy célváltozó egy adott értéke valószínűségének kiszámítása¹.



1.4. ábra. Következtetés polifában: az X célváltozó egymástól független részekre szeparálja a teljes hálót.

A feladat tehát a $P(X|E)$ valószínűség kiszámítása, amely, ha az evidenciák teljes halmazát kettébontjuk az X „feletti” és X „alatti” részekre, a Bayes-szabály segítségével a következő alakra hozható:

$$P(X|E) = P(X|E_X^+, E_X^-) = \frac{P(E_X^-|X, E_X^+)P(X|E_X^+)}{P(E_X^-|E_X^+)}. \quad (1.8)$$

¹Mind a teljes marginális eloszlás, mind egy többváltozós konfiguráció valószínűségének kiszámítása megoldható hasonló nagyságrendben, hisz a marginális kiszámítható az eljárás egyszerű megismétlésével a változó összes értékére, egy többváltozós konfiguráció esetén pedig alkalmazható a láncszabály: kiszámítjuk az első célváltozó valószínűségét, ezután felvesszük az evidenciák közé, és kiszámítjuk a második célváltozó valószínűségét az új, bővített evidencia mellett, majd végül a kapott valószínűségeket összeszorozva kapjuk az eredeti lekérdezésre a választ. Vagyis a következtetés nagyságrendje $\mathcal{O}(cn)$, illetve $\mathcal{O}(n^2)$ nagyságrendben, ahol c az adott változó kardinalitása, n pedig a változók száma

A $P(E_X^-|E_X^+)$ értéke kezelhető konstansként, így elhagyható, hiszen az X feletti eloszlást majd X értékei felett kapott értékek normalizálásaként kapjuk; továbbá, mivel X d-szeparálja E_X^+ -t és E_X^- -t az egyenlet a következőképpen egyszerűsíthető:

$$P(X|E) \propto P(E_X^-|X)P(X|E_X^+). \quad (1.9)$$

$P(X|E_X^+)$ kiszámítása A $P(X|E_X^+)$ mennyiséget egyszerűen kiszámíthatnánk X szüleinek (\mathbf{U}) eloszlásai ismeretében, a $P(X|U)$ feltételes valószínűségek $P(U)$ szerint súlyozott összegeként. Ezt felhasználva $P(X|E_X^+)$ tovább bontható:

$$P(X|E_X^+) = \sum_u P(X|u, E_X^+)P(u|E_X^+). \quad (1.10)$$

Mivel X d-szeparálja \mathbf{U} elemeit, és E_X^+ azokhoz tartozó részeit (az 1.4 ábrán E -nek a bejelölt dobozokban lévő részeit), $P(u|E_X^+)$ felírható a $P(u_i|E_X^+)$ mennyiségek szorzataként; a $P(u_i|E_X^+)$ valószínűségekben pedig E_X^+ a szűkebb $E_{U_i \setminus X}$ halmazokra cserélhető:

$$P(X|E_X^+) = \sum_u P(X|u) \prod_i P(u_i|E_{U_i \setminus X}). \quad (1.11)$$

Ez a képlet már alkalmas lesz arra, hogy az eredetibe visszaírva egy rekurzív eljárás alapját képezze, hisz $P(X|u)$ -k a Bayes-háló CPT-inek bejegyzései, $P(u_i|E_{U_i \setminus X})$ pedig $(P(X|E))$ -hez hasonlóan számítható, viszont ez már kevesebb evidenciát tartalmaz vagyis intuitíve már látható, hogy az algoritmus ezen része terminálni fog.

$P(E_X^-|X)$ számítása Kihhasználva a háló struktúrájának fagráf voltát (hasonlóképpen az 1.11 egyenletre vezető megfontolásokhoz), $P(E_X^-|X)$ is felírható egy X gyermekei (Y_i) feletti szorzatként:

$$P(E_X^-|X) = \prod_i P(E_{Y_i \setminus X}|X), \quad (1.12)$$

amelynek a tagjait ha felírjuk X gyermekei (Y_i) és azok szülei (\mathbf{Z}_i) feletti összegzésekként, kapjuk:

$$P(E_X^-|X) = \prod_i \sum_{y_i} \sum_{z_i} P(E_{Y_i \setminus X}|X, y_i, z_i)P(y_i, z_i|X). \quad (1.13)$$

Ha $E_{Y_i \setminus X}$ -et E -hez hasonlóan felbontjuk Y_i alatti és feletti részekre ($E_{Y_i}^-$ és $E_{Y_i \setminus X}^+$):

$$P(E_X^-|X) = \prod_i \sum_{y_i} \sum_{z_i} P(E_{Y_i}^-|X, y_i, z_i)P(E_{Y_i \setminus X}^+|X, y_i, z_i)P(y_i, z_i|X). \quad (1.14)$$

Kihhasználva, hogy Y_i d-szeparálja $E_{Y_i}^-$ -t X -től és z_i -től, illetve, hogy z_i d-szeparálja $E_{Y_i \setminus X}^+$ -t X -től és y_i -től, valamint kiemelve egy z_i -től nem függő tagot a második szummából:

$$P(E_X^-|X) = \prod_i \sum_{y_i} P(E_{Y_i}^-|y_i) \sum_{z_i} P(E_{Y_i \setminus X}^+|z_i)P(y_i, z_i|X), \quad (1.15)$$

majd alkalmazva a Bayes-szabályt $P(E_{Y_i \setminus X}^+ | z_i)$ -re:

$$P(E_X^- | X) = \prod_i \sum_{y_i} P(E_{Y_i}^- | y_i) \sum_{z_i} \frac{P(z_i | E_{Y_i \setminus X}^+) P(E_{Y_i \setminus X}^+)}{P(z_i)} P(y_i, z_i | X), \quad (1.16)$$

amiből:

$$P(E_X^- | X) = \prod_i \sum_{y_i} P(E_{Y_i}^- | y_i) \sum_{z_i} \frac{P(z_i | E_{Y_i \setminus X}^+) P(E_{Y_i \setminus X}^+)}{P(z_i)} P(y_i | X, z_i) P(z_i | X). \quad (1.17)$$

Itt kihasználhatjuk, hogy a d-szeparáció miatt $P(z_i | X) = P(z_i)$, illetve $P(E_{Y_i \setminus X}^+)$ -t átnevezve β_i normalizációs állandóvá:

$$P(E_X^- | X) = \prod_i \sum_{y_i} P(E_{Y_i}^- | y_i) \sum_{z_i} \beta_i P(z_i | E_{Y_i \setminus X}^+) P(y_i | X, z_i). \quad (1.18)$$

A β_i -k kiemelhetők az összegzések elé, az így keletkező $\prod_i \beta_i$ pedig elhagyható a későbbi normalizáció miatt, Y_i szülei (Z_{ij} -k) pedig függetlenek lesznek egymástól Y_i ismeretében, így együttes valószínűségük szorzatként felírható:

$$P(E_X^- | X) \propto \prod_i \sum_{y_i} P(E_{Y_i}^- | y_i) \sum_{z_i} P(y_i | X, z_i) \prod_j P(z_{ij} | E_{Z_{ij} \setminus Y_i}). \quad (1.19)$$

A fenti kifejezés már szintén felhasználható a rekurzív algoritmusban, hisz

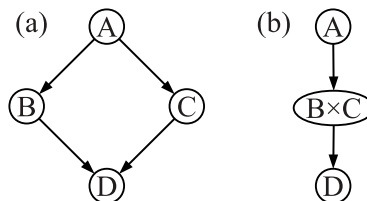
- $P(E_{Y_i}^- | y_i)$ a rekurzív megjelenése $P(E_X^- | X)$ -nek;
- $P(y_i | X, z_i)$ CPT bejegyzések a hálóban;
- $P(z_{ij} | E_{Z_{ij} \setminus Y_i})$ pedig a rekurzív megjelenése $P(X | E)$ -nek.

Az eljárás komplexitása. Látható, hogy az eljárás minden csomópontot maximum egyszer látogat meg, hiszen az a célcsomópontból indított rekurzív hívásokból áll, amelyek az élek mentén haladnak a célcsomóponttól távolodva (a rekurzív hívások során az azt indító előző csomópont kizáródik a meglátogatott csomópontok köréből); ebből pedig következik hogy komplexitása arányos a csomópontok számában lineáris.

1.3.4. Következtetés többszörösen összekötött hálóknban

Ha a hálóban több út is vezet két csomópont között, a fenti algoritmus nem alkalmazható. Ilyen helyzetben alkalmazhatunk egy az összekötöttség fokára érzéketlen egzakt algoritmust, mint például a felsorolásos (1.3.1 fejezet) vagy a változóeliminációs (1.3.2 fejezet) eljárást, de ezek számítási igénye praktikusán csak játék-alkalmazásokban elfogadható. A hálót átalakíthatjuk egy polifa struktúrába valamilyen eljárással. Ebben az alfejezetben ilyen eljárásokra mutatunk egyszerűbb példákat, illetve az 1.4 alfejezet fog bemutatni egy összetettebbet. Egy további lehetőség, hogy lemondunk az egzakt következtetésről és valamilyen sztochasztikus szimulációs (Monte Carlo) eljárást alkalmazunk. Ezek bemutatásával és háttérükkel az 1.5 alfejezet foglalkozik.

Összevonós eljárások A háló csomópontjai közül bizonyosakat összevonunk ú.n. megacsomópontokba amelyek az általuk tartalmazott eredeti csomópontok együttes feltételes eloszlását tartalmazzák. Az 1.5 ábra egy ilyen esetet illusztrál.



1.5. ábra. Többszörösen összekötött háló és a belőle összevonással létrehozott, megacsomópontokat tartalmazó transzformált háló.

Bár a következtetés feladata ebben az új struktúrában már végrehajtható a lineáris időben, az eredeti háléhoz viszonyítva ez nem áll fenn, hisz az összevont csomópontok értékészlete a tartalmazott csomópontok értékészletének Descartes-szorzata.

Feltételezéses eljárások A *feltételezéses eljárások* az előző megközelítésnek éppen az ellenkezőjét alkalmazzák, itt az egyetlen, de (az egyes csomópontok szintjén) bonyolultabb másodlagos struktúra helyett több, de egyszerűbb hálót hozunk létre, amelyekben már egyszerűen végrehajtható a következtetés. Az itt bemutatott *vágóhalmaz feltételezésen alapuló eljárás* a következő fő lépésekből áll:

- Keressük meg a csomópontok egy olyan halmazát, amely konfigurációját rögzítve a háló struktúrája egy fává redukálódik (az ilyen csomópont-halmazt nevezzük *vágóhalmaznak*, jelöljük ezt C -vel).
- A vágóhalmaz lehetséges konfigurációival származtatott hálók mindegyikén végezzük el a következtetést: $P(X|E \cup \{C = c\})$.
- A keresett valószínűség a fenti valószínűségeknek a vágóhalmaz egyes konfigurációinak valószínűségével súlyozott átlagaként adódik:

$$P(X|E) = \sum_{\forall c} P(X|E \cup \{C = c\})P(C = c|E) \quad (1.20)$$

Ennél az eljárásnál is látható, hogy a következtetés exponenciális volta nem kerülhető meg: általános esetben a kiértékelendő fák száma a vágóhalmaz méretében exponenciális (egészen pontosan a vágóhalmaz elemei értékészletei Descartes-szorzatának számossága).

Adott pontosságú vágóhalmaz feltételezés. Ennél a módszernél lehetőség van a számítási komplexitás csökkentésére –természetesen a pontosság rontása, és az egzaktságról való lemondás árán–: megtehetjük, hogy csak adott összvalószínűségű alhálóra végezzük el

a következtetést és az ezekből adódó eredményt korrigáljuk a figyelembe vett hálók összvalószínűségével. Ha tehát a válasz pontosságának például 0,1-en belül kell lennie, annyi alhálót kell figyelembe vennünk, amennyinek az összvalószínűsége a 0,9-et meghaladja.

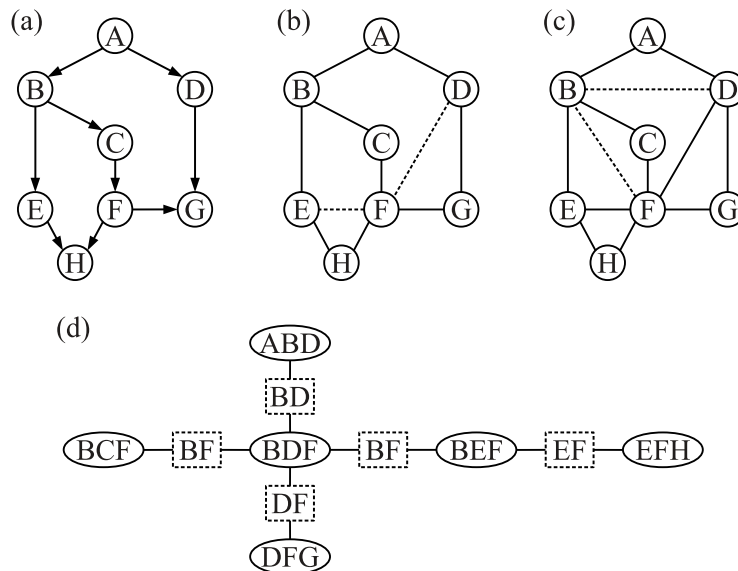
1.4. A PPTC következtetés

A PPTC (*Probability Propagation in Trees of Clusters*) eljárás során az eredeti Bayes-hálóból egy másodlagos struktúrát hozunk létre, amely a változók eloszlását (illetve potenciálisan az evidenciákat is) már egy olyan formában tárolja, amely egyszerű eljárásokkal, közvetlenül felhasználható a lekérdezések megválaszolására. Mint azt a továbbiakban összefoglaljuk, az eljárás két fő részből áll: először a gráfstruktúrát transzformáljuk egy ún. *klikkfába*; majd ezután az eredeti háló csomópontjainak feltételes eloszlásait, illetve az evidenciákat visszük be a klikkfa csomópontjaiba.

Az eljárást először Lauritzen és Spiegelhalter publikálták [2], ebben a fejezetben pedig Huang és Darwiche metodológiai cikkének [1] főbb elemeit ismertetjük.

1.4.1. Klikkfa konstruálása

A PPTC eljárás első részeként az eredeti struktúrából egy másodlagos struktúrát hozunk létre. A következő részekben ennek a lépéseit vesszük sorra. Az 1.6 ábra ennek a folyamatnak a lépéseit illusztrálja.



1.6. ábra. A másodlagos struktúra létrehozásának lépései a PPTC algoritmusban: az eredeti Bayes-háló struktúra (a), a morális gráf (b), a háromszögesített gráf (c), a klikkeket (folytonos) és szeparáló halmazokat (szaggatott) tartalmazó klikkfa (d).

Morális gráf formára alakítás Első lépésként a DAG-ot annak morális gráfjává alakítjuk, azaz összekötjük az egyes csomópontok szüleit egymással, majd töröljük az élek irányítottságát.

Háromszögesített gráf A kapott morális gráfot *háromszögesíteni* kell, azaz minden, 3-nál hosszabb körben kell lennie 2, a körben nem szomszédos, összekötött csomópontnak. A háromszögesítés megvalósítására az 1.2 algoritmus szolgálhat.

Algorithm 1.2 Morális gráf G_M háromszögesítése

- 1: Készítsünk G_M -ről egy másolatot: G'_M
 - 2: **while** $G'_M \neq \emptyset$ **do**
 - 3: Válasszuk ki G'_M -ből a V csúcsot a 8. sortól leírt kritérium szerint.
 - 4: V és szomszédai egy ú.n. *klikket* alkotnak, kössük össze az összes csomópontpárt a klikkben, és az újonnan hozzáadott éleket húzzuk be G_M -ben is.
 - 5: Töröljük V -t G'_M -ből.
 - 6: **end while**
 - 7: G_M az újonnan hozzáadott élekkel így már háromszögesítve van.
-
- 8: A 3. sorban alkalmazott kritérium:
 - 9: Egy csomópont *súlya* annak értékészletének számossága.
 - 10: G'_M csomópontjai közül válasszuk azt, amelyik a 4 sorban a lehető legkevesebb él gráfhoz adását indukálja.
 - 11: Ha több ilyen is van, válasszuk a legkisebb súlyút.
-

Mint látható, G_M hatékony háromszögesítéséhez minden csomóponthoz két számértéket kell számon tartani: az általa esetlegesen okozandó él-hozzáadások számát, és a hozzá tartozó klikk súlyát. Ezeket az értékeket azonban nem kell minden csomópont-törléskor a teljes hálóra újraszámolni, elég csak az utoljára törölt V csomópont szomszédaira újraszámolni őket.

Klikkek azonosítása A háromszögesített gráfban azonosítani kell a maximális klikkeket (teljesen összekötött csomóponthalmazokat), ezek fogják a klikkfa csomópontjait alkotni. Bár vannak általános eljárások tetszőleges gráfban való klikk-keresésre, ebben a lépésben felhasználhatjuk, hogy (1) minden a gráfban lévő klikk a háromszögesítés során jött létre; és (2) egy ilyen klikk nem lehet egy korábban létrehozott másik klikk részhalmaza. Ebből következően, elég, ha a háromszögesítés során feljegyezzük azokat a klikkeket, amelyek az eddig lementettek egyikének sem részhalmazai: ez a lista meg fog egyezni a klikkekével

Optimális klikkfa építése A fentebb előállított klikkekből ezután ki kell alakítani a klikkfát, mégpedig úgy, hogy lépésről-lépésre összekötögetjük a klikkeket, amíg a fa az összeset nem tartalmazza. A klikkek összekötése formálisan úgy történik, hogy közéjük a gráfban egy ú.n. *szeparáló halmazt* (angol terminológiával *sepset-et*) illesztünk a gráfban.

A seplet ugyanúgy az eredeti háló csomópontjainak egy halmazát tartalmazza, mint a klikkek, nevezetesen azokat, amelyek a hozzá kapcsolódó mindkét klikkben jelen vannak.

Az, hogy a klikkeket milyen módon kötjük össze egy fában, a teljesítmény miatt lényeges. Az 1.3 algoritmus egy olyan kiválasztási kritériumot tartalmaz, amely biztosítja, hogy a klikkfa a következtetés számítási komplexitása szempontjából optimális legyen.

Algorithm 1.3 Optimális klikkfa építése

- 1: Kezdjük n darab különálló, egy csomópontos gráffal (a $c \in \mathcal{C}$ klikkekből) és egy \mathcal{S} üres halmazzal.
 - 2: **for each** (X, Y) **in** $(X, Y) : X, Y \in \mathcal{C}, X \neq Y$ **do**
 - 3: Állítsuk elő az S_{XY} seplet-et, amely X -re és Y -ra hivatkozik szomszédaként és $X \cap Y$ csomópontokat tartalmazza.
 - 4: Adjuk hozzá S_{XY} -t \mathcal{S} -hez.
 - 5: **end for each**
 - 6: **repeat**
 - 7: Válasszunk ki egy S_{XY} seplet-et \mathcal{S} -ből, a 11. sortól kezdődően leírt kritérium szerint.
 - 8: Töröljük S_{XY} -t \mathcal{S} -ből.
 - 9: Illesszük be S_{XY} -t az általa hivatkozott X és Y klikkek közé, feltéve, hogy X és Y külön fában vannak.
 - 10: **until** Beillesztettünk $n - 1$ seplet-et.
-
- 11: A 7. sorban alkalmazott kritérium.
 - 12: S_{XY} seplet *tömege* az általa tartalmazott csomópontok száma, azaz $|X \cap Y|$
 - 13: S_{XY} seplet *költsége* X és Y klikkek súlyának összege, ahol egy klikk *súlya* az őt alkotó csomópontok súlyainak szorzata (lásd 1.2 algoritmus, 9. sor).
 - 14: Az $S_{XY} \in \mathcal{S}$ seplet-ek közül válasszuk azt, amelyiknek a legnagyobb a tömege.
 - 15: Ha több ilyen is van, akkor ezek közül azt, amelyiknek a legkisebb a költsége.
-

Üres seplet-ek A fenti algoritmusban olyan seplet-ek is előfordulhattak, amelyek nem tartalmaztak egy csomópontot sem. Bár a legtöbb esetben ezek a seplet-ek nem jelennek meg a végső struktúrában, ha az eredeti háló nem volt teljesen összekötve, a klikkfa is tartalmazni fog ilyen seplet-et (egészen pontosan annyit, ahány a különálló komponensek összekötéséhez kell, vagyis ha az eredeti gráf K darab komponensből állt, akkor $K - 1$ -et).

1.4.2. Valószínűségek terjesztése a klikkfában

Ha a fenti módon előállt a klikkfa struktúrája, a klikkek és seplet-ek által tárolt valószínűségi eloszlásokba be kell vinni az eredeti Bayes-háló által reprezentált értékeket. Ez a következő fő lépésekkel történik.

Inicializáció A csomópontok (vagyis klikkek és sepek) által tárolt valószínűségi potenciálok inicializálása az 1.4 algoritmus szerint történik.

Algorithm 1.4 Klikkfa potenciáljának inicializálása

```

1: for each  $X$  in  $\mathcal{C} \cup \mathcal{S}$  do
2:    $\phi_X \leftarrow 1$ 
3: end for each
4: for each  $V$  in  $U$  do
5:   Keressünk egy  $X$  klikket, hogy  $V \cup Pa(V) \subseteq X$ 
6:    $\phi_X \leftarrow \phi_X P(V|Pa(V))$ 
7: end for each

```

Vagyis először minden klikk és sepek potenciálját „feltöltjük” 1-esekkel, majd pedig minden BN csomóponthoz keresünk egy olyan klikket, amely tartalmazza annak családját, és annak potenciáljába pontonkénti szorzással bevisszük a változó feltételes valószínűségi eloszlását, $P(V|Pa(V))$ -t. Ezután a klikkfára igaz lesz a következő:

$$\frac{\prod_{i=1}^N \phi_{X_i}}{\prod_{j=1}^{N-1} \phi_{S_j}} = \frac{\prod_{k=1}^Q P(V_k | \mathbf{C}_{V_k})}{1} = P(U) \quad (1.21)$$

(ahol N a klikkek, Q a BN csomópontjainak száma, ϕ_H pedig a H halmaz feletti potenciál), vagyis a klikkfa globálisan konzisztens lesz.

Evidencia bevitele. Ha bizonyos változókról rendelkezésre állnak evidenciák is, akkor ezek is könnyen bevihetők a rendszerbe: ha λ_V jelöli a V változóval kapcsolatos ismereteinket leíró valószínűségi potenciált, akkor elég egy megfelelő (azaz V -t tartalmazó) X klikkbe bevinni ezt is egy pontonkénti szorzással:

$$\phi_X \leftarrow \phi_X \lambda_V \quad (1.22)$$

A valószínűségek globális terjesztése. A valószínűség-terjesztés célja, hogy a klikkfát lokálisan is konzisztenssé tegye, vagyis annak csomópontjai valóban az általuk tartalmazott változók együttes eloszlását tartalmazzák. Ezt a műveletet egy sor, szomszédos klikkek közötti ún. üzenetküldéssel valósíthatjuk meg. Egy X -ből S_{XY} -on keresztül Y -ba küldött üzenet után R_{XY} konzisztens lesz X -szel (vagyis $\phi_{S_{XY}} = \sum_{X \setminus (X \cap Y)} \phi_X$), miközben a $P(U) = \frac{\prod_i \phi_{X_i}}{\prod_j \phi_{S_j}}$ egyenlet sem sérül.

A globális valószínűség-terjesztés során minden klikk minden egyes szomszédjához pontosan egy üzenetet fog küldeni, és a teljes folyamat végére a klikkfa lokálisan konzisztens lesz. Egyetlen üzenetküldés a következők szerint történik, az alábbi két lépésben:

Projekció. Tároljuk el S_{XY} régi tábláját, és rendeljünk hozzá egy újat:

$$\phi'_{S_{XY}} \leftarrow \phi_{S_{XY}} \quad (1.23)$$

$$\phi_{S_{XY}} \leftarrow \sum_{X \setminus S_{XY}} \phi_X \quad (1.24)$$

Abszorpció. Rendeljünk Y -hoz egy új táblát S_{XY} régi és új táblájának felhasználásával:

$$\phi_Y \leftarrow \phi_Y \frac{\phi_{S_{XY}}}{\phi'_{S_{XY}}}. \quad (1.25)$$

Belátható [Jensen], hogy $\phi'_{S_{XY}}$ csak ott vehet fel 0 értéket, ahol $\phi_{S_{XY}}$ is, az ilyen esetekben vegyük úgy, hogy $\frac{0}{0} = 0$

A teljes valószínűség terjesztés során egy tetszőlegesen választott X klikkből indulunk, majd ebből kiindulva történik meg a $2(n-1)$ darab üzenetküldés. Ez két fázisban történik, az első, ú.n. bizonyíték-gyűjtési fázisban minden klikk X irányába küld üzenetet, a második, ú.n. bizonyíték-elosztási fázisban (X -szel kezdve) minden klikk X -től távolabbi szomszédainak küld üzenetet, az 1.5 rekurzív algoritmus szerint. Ebben minden egyes klikkhez egy jelzőbitet ($b(X)$) rendelünk, amelynek segítségével számon tartjuk, hogy az adott fázisban mely klikkek küldtek már üzenetet és melyek nem.

Normalizálás. Ha a fentiek során bizonyítékok bevitelére is sor került az egyes csomópontok nem feltétel nélküli, hanem az evidenciákkal együtt vett eloszlásokat fognak tartalmazni, vagyis például $P(V, e)$ -t. Ahhoz, hogy ezekből megkapjuk az általunk keresett $P(V|e)$ -ket szükséges még a klikkek és csomópontok tábláinak normalizálása is.

1.4.3. Következtetési esetek

Ha a fenti módon a klikkfát lokálisan és globálisan is konzisztenssé tettük, végre elkezdhetünk benne következtetni. Az alábbiakban röviden vázoljuk, hogy az alapvető következtetési esetek megoldását.

Egyetlen változó marginálisa. A lokális konzisztenciának megfelelően minden klikk és sepsét táblája egy valós többváltozós marginálist tartalmaz, ennek megfelelően, ha egyetlen változóra vagyunk kíváncsiak, elég keresnünk egy (lehetőleg minél kevesebb változót tartalmazó) csomópontot, és annak táblájából marginalizálással előállíthatjuk a keresett valószínűségi eloszlást.

Több változó marginálisa. Ha létezik olyan klikkfa csomópont, amely tartalmazza az összes lekérdezés-változót, a marginalizálás itt is végrehajtható. Ha viszont nincs olyan klikk, amely teljes egészében lefedné a célváltozók halmazát, a láncszabályt kell alkalmaznunk.

Egy másik lehetőség még a globális konzisztencia, vagyis a

$$P(U) = \frac{\prod_i \phi_{X_i}}{\prod_j \phi_{S_j}} \quad (1.26)$$

egyenlet kihasználása. Ezt felhasználva a keresett valószínűségek szintén előállíthatók (bár a szerző tapasztalata szerint jóval kevésbé hatékonyan, mint a láncszabály alkalmazásával).

Algorithm 1.5 Globális valószínűség-terjesztés

```

1: Válasszunk egy tetszőleges klikket:  $X$ .

2: for each  $C$  in  $\mathcal{C}$  do  $b(C) \leftarrow \text{false}$ 
3: end for each
4: call COLLECT-EVIDENCE( $X$ )

5: for each  $C$  in  $\mathcal{C}$  do  $b(C) \leftarrow \text{false}$ 
6: end for each
7: call DISTRIBUTE-EVIDENCE( $X$ )



---


8: function COLLECT-EVIDENCE(klikk  $X$ )
9:    $b(X) \leftarrow \text{true}$ 
10:  for each  $Y$  in  $X$  szomszédai do
11:    if  $b(Y) = \text{false}$  then call COLLECT-EVIDENCE( $Y$ )
12:    end if
13:  end for each
14:  Küldjünk üzenetet  $X$ -ből abba a csomópontba, amely ezt a függvényhívást indította.
15: end function



---


16: function DISTRIBUTE-EVIDENCE(klikk  $X$ )
17:    $b(X) \leftarrow \text{true}$ 
18:  for each  $Y$  in  $X$  szomszédai do
19:    if  $b(Y) = \text{false}$  then
20:      Küldjünk üzenetet  $X$ -ből  $Y$ -ba.
21:      call COLLECT-EVIDENCE( $Y$ )
22:    end if
23:  end for each
24: end function

```

1.5. Közelítő következtetés sztochasztikus szimulációval

Sztochasztikus szimulációs eljárás alkalmazására akkor van szükség, amikor az egzakt következtetési eljárások –azok számítási költségei miatt– nem alkalmazhatók. A Monte Carlo eljárások alapfeladata és a hozzá tartozó alapelve a következő: adott egy valószínűségi változótól függő mennyiség, amelynek keressük a valószínűségi változó eloszlása szerinti várhatóértékét; mivel a teljes eloszlás nem kezelhető, ezért mintavételezzük azt, és a mintából számított statisztikával becsüljük a keresett mennyiséget. Képlettel kifejezve:

$$E_{P(X)}(f(X)) \approx \hat{E}_{P(X)}(f(X)) = \frac{1}{N} \sum_{i=1}^N f(X_i). \quad (1.27)$$

A mi konkrét esetünkben, a Bayes-hálókból való prediktív következtetés esetén, a mintavételezendő eloszlás a háló által reprezentált együttes eloszlás feltéve a bizonyítékokat, a keresett mennyiség pedig a célváltozó(k) marginális eloszlása. Látható, hogy az 1.3.1 alfejezetben bemutatott felsorolásos módszer a mintavételezendő eloszlás kimerítő felsorolásos feldolgozásának felel meg.

Mivel azonban ez a naiv megközelítés a legritkább esetekben vezet csak eredményre, ebben az alfejezetben először olyan egyszerűbb eljárásokat mutatunk be, amelyek magát a háló eloszlását mintavételezik; majd a *Markov-láncos Monte Carlo* (Markov Chain Monte Carlo – MCMC) módszerek rövid elméleti áttekintése és a Bayes-hálós következtetésre való alkalmazásának bemutatása következik.

1.5.1. Mintagenerálás „üres” hálóból

A mintavételezésen alapuló következtetési eljárások mindegyike azon alapul, hogy a háló (és esetleg az evidenciák) alapján generál egy mintahalmazt, amely alapján már kiszámíthatunk egy statisztikát, amellyel a kérdéses valószínűséget becsülhetjük. A legegyszerűbb esetben az evidenciákat figyelmen kívül hagyva generálunk egy mintahalmazt a háló által leírt együttes eloszlásból.

Egyetlen minta generálása Bayes-hálóból Az eljárás alapeleme egyetlen minta generálása, amely a következő lépésekből áll:

1. Vegyük a csomópontok egy topologikus sorrendjét, kezdetben minden csomópont álljon behelyettesített érték nélkül. Ebben a sorrendben minden egyes csomópontra:
2. Sorsoljunk ki egy értéket az aktuális csomópontnak annak a háló által leírt feltételes valószínűségi eloszlásából, feltéve a szülők aktuális értékeit ($P(X|Pa(X))$); adjuk értékül a csomópontnak a kisorsolt értéket.
3. Ha végiglépdeltünk az összes csomóponton, a nekik adott értékek alkotják a kisorsolt konfigurációt.

1.5.2. Elutasító mintavételezés

A fenti eljárás kellő számú ismétlésével kapott minta már egyszerűen felhasználható egy adott (hálóbeli) esemény adott evidenciák melletti valószínűségének megbecslésére: ez egyszerűen az céleseménnyel és az evidenciákkal is kompatibilis minták számának aránya lesz az evidenciákkal kompatibilis esetek számához.

Az ún. *elutasító mintavételezés* a fentieknek megfelelően még két lépést alkalmaz a generált mintán:

1. Kiszűri („elutasítja”) a mintából azokat, amelyek nem kompatibilisek az evidenciákkal.

2. A fennmaradó mintán a kérdéses feltételes valószínűség a célkonfigurációval kompatibilis minták számának aránya a teljes (már megszűrt) mintaszámhoz.

A módszer fő előnye az egyszerűsége, hátránya viszont, hogy a becslés során valóban felhasználható minták száma az evidenciák mennyiségével exponenciálisan csökken.

1.5.3. Valószínűségi súlyozás

Ez az algoritmus az előző módosítása annak az érdekében, hogy elkerülhessük a becslésben fel nem használható (az evidenciákkal inkompatibilis) minták generálását. Ennek eléréséhez az alap mintagenerálási lépést annyiban módosítjuk, hogy az evidencia-változók értékeit rögzítjük és csak a fennmaradóknak sorsolunk; az így nyert mintákat azonban súlyoznunk kell: míg korábban minden egyes minta 1.0 súllyal szerepelt, most ezt a súlyt meg kell szorozni az evidencia-változók feltételes valószínűségeivel (intuitíve látható, hogy ezzel lényegében a mintavételezés során elutasított –azaz az evidenciákkal inkompatibilis– mintákkal együtt kapott számokhoz képesti arányokat kapjuk meg).

A becsült valószínűség maga az előző eljáráshoz hasonlóan, a lekérdezéssel kompatibilis és az összes minták összsúlyainak arányaként áll elő.

Bár ez az eljárás már hatékonyabb, mint a *elutasító mintavételezés*, mert figyelembe veszi az összes legenerált mintát, az még mindig problémát jelenthet (hatékonysági szempontból), hogy előfordulhat, hogy a teljes minta összsúlyának egy nagyon nagy részét a minták egy kis hányada képviseli. Azok a minták ugyanis, amelyek valószínűségeileg kevésbé illeszkednek csak a bizonyítékékhöz kis súlyokat kapnak így hatásuk a végső becslés szempontjából sem lesz jelentős.

A továbbiakban bemutatott Gibbs-mintavételező ezt a gyengeséget orvosolja oly módon, hogy a mintavételezés során végig olyan konfigurációkat sorsol, amelyek az adott szituációban valószínűek. Mielőtt a Gibbs-mintavételező részletezésére térnénk, áttekintjük az annak alapjául szolgáló Monte Carlo-eljárásokat.

1.6. A Monte Carlo-eljárások áttekintése

A Monte Carlo eljárások alapötlete, hogy egy eloszlás feletti integrált vagy (az ebben a fejezetben vizsgált diszkrét esetben) összegzést annak analitikus megoldása helyett valamilyen mintavételezési eljárás segítségével közelítsenek.

Adott tehát a kiszámítandó

$$\bar{f} = E_{\pi(X)}[f(X)] \quad (1.28)$$

mennyiség, amelynek közelítésére a következő lépéseket alkalmazzuk:

- Generáljunk egy $\{X_i\}_{i=1}^N$ független, azonos eloszlású (*independent, identically distributed – i.i.d.*) mintahalmazt $\pi(X)$ mintavételezésével.
- A minta alapján számoljuk ki az $\hat{f}_N = \sum_{i=1}^N f(X_i)$ közelítést.

- Adjunk valamiféle megbízhatósági becslést az $|\bar{f} - \hat{f}_N|$ valós és becslt érték közötti eltérésre.

Bayes-hálókbán való következtetés esetén a mintavételezendő $\pi(X)$ eloszlás a háló által reprezentált $P(\mathcal{U})$ együttes eloszlás, esetleg ennek az evidenciák szerinti feltételes értéke: $P(\mathcal{U}|E)$. A keresett \bar{f} érték pedig a célváltozók marginális eloszlása, esetleg azok egy konfigurációjának valószínűsége.

A becslés megbízhatóságáról a következő eredmények léteznek:

- A *nagy számok erős törvénye* alapján az \hat{f}_N becslés erősen konzisztens, azaz:

$$P(\lim_{N \rightarrow \infty} \hat{f}_N = \bar{f}) = 1. \quad (1.29)$$

- A *központi határeloszlás tétele* szerint \hat{f}_N standardizáltja aszimptotikusan Gauss-eloszlású lesz:

$$\frac{\hat{f}_N - \bar{f}}{\sigma_N} \rightarrow N(0, 1), \text{ ha } N \rightarrow \infty, \text{ ahol } \sigma_N = \frac{\text{Var}(f(X))}{\sqrt{N}} \quad (1.30)$$

- Ha, mint a mi esetünkben, $f(X)$ korlátos, akkor a közelítés konvergenciájának sebességére további becslések adhatók a Hoeffding- és a Bernstein-egyenlőtlenségek alapján (specifikusan, ha $0 \leq f(X) \leq 1$):

$$p(|\hat{f}_N - \bar{f}| \geq \epsilon) \leq 2 \exp(-2\epsilon^2 N) \leq \delta?? \quad (1.31)$$

$$E[|\hat{f}_N - \bar{f}|] \leq \sqrt{\frac{c_0}{N}} \quad (1.32)$$

1.6.1. Fontossági mintavételezés

Abban az esetben, ha a $\pi(X)$ eloszlás nem mintavételezhető hatékonyan, de rendelkezésre áll egy megegyező tartójú $q(X)$ eloszlás, amelyen ez megtehető, alkalmazhatjuk a fontossági mintavételezést. Ebben az esetben a $q(X)$ -ből vett mintákat a következő képleten keresztül használhatjuk fel \bar{f} közelítésére:

$$\bar{f} = E_{\pi(X)}[f(X)] = E_{q(X)}\left[\frac{f(X)\pi(X)}{q(X)}\right] \quad (1.33)$$

vagyis az \hat{f}_N közelítés az alábbi szerint számítható:

$$\hat{f}_N = \frac{\frac{1}{N} \sum_{t=1}^N w(X_t) f(X_t)}{\frac{1}{N} \sum_{t=1}^N w(X_t)} = \frac{1}{N} \sum_{t=1}^N w^*(X_t) f(X_t), \quad (1.34)$$

ahol $w(X_t)$ és $w^*(X_t)$ a fontossági súlyok:

$$w(X_t) = \frac{\pi(X_t)}{q(X_t)}, w^* = \frac{w(X_t)}{\frac{1}{N} \sum_{t=1}^N w(X_t)} \quad (1.35)$$

1.6.2. Markov-láncok

1.3. definíció (Markov-lánc). Az $\mathcal{X} = \{X_0, X_1, \dots\}$ valószínűségi változók sorozata egy (elsőrendű) Markov-lánc, ha $P(X_t | X_{t-1}, \dots, X_0) = P(X_t | X_{t-1})$. A Markov-lánc homogén, ha $P(X_t | X_{t-1})$ nem függ t -től.

A fenti X_t változókat általában a Markov-lánc *állapotainak* tekintjük, amelyeket az egymás után felvesz; ahhoz, hogy egy folyamat Markov-lánc lehessen –mint az a definícióból látszik– az kell, hogy állapotai a múltbeli állapotoktól csak a megelőzőn keresztül függjenek. A t paraméternek gyakran valamiféle időbeli interpretációt tulajdonítunk.

A továbbiakban az X_t állapotokat természetes számokkal (i, j, \dots) jelöljük, és mivel feltesszük, hogy a vizsgált Markov-láncok homogének (vagyis idő-invariánsok), az i állapotból j -be való átlépés valószínűségét $p_{ij} = P(X_t = j | X_{t-1} = i)$ -vel jelöljük. A p_{ij} értékekből alkotott *egylépéses állapot-átmeneti mátrix* pedig $P = [p_{ij}]$. Az n -lépéses állapot-átmeneti mátrix P n -edik hatványa: $P^{(n)} = P^n$.

1.4. definíció (Invariáns eloszlás). A p^{inv} eloszlást a χ homogén Markov-lánc invariáns eloszlásának nevezzük, ha $p^{inv} = p^{inv} P$, ahol P χ állapotátmenet-mátrixa. (Következésképpen, ha $p^{(0)} = p^{inv}$, akkor $\forall t : p^{(t)} = p^{inv}$.)

Markov-láncok tulajdonságai

Az alábbiakban sorra vesszük a Markov-láncok legfontosabb tulajdonságait, és a hozzájuk kapcsolódó alapvető tételket.

1.5. definíció (Stabilitás). Az \mathcal{X} Markov-lánc stabil, ha a $\lim_{t \rightarrow \infty} p(X_t) = p^{(\infty)}$ létezik, valóban egy eloszlás és független a $p(X_0)$ kiindulási eloszlástól. $p^{(\infty)}$ -t határérték- vagy egyensúlyi eloszlásnak nevezzük.

1.6. definíció (Irreducibilitás). A diszkrét, véges állapotú \mathcal{X} Markov-lánc irreducibilis, ha bármely (i, j) párra létezik $n_{ij} > 0$, hogy $p_{ij}^{(n_{ij})} > 0$.

1.7. definíció (Aperiodicitás). A diszkrét, véges állapotú \mathcal{X} Markov-lánc aperiodikus, ha van olyan i (irreducibilitást feltételezve ez bármely i -re igaz), hogy létezik $n_i > 0$, hogy bármely $n \geq n_i$ -re $p_{ii}^{(n)} > 0$

1.3. tétel. Ha egy diszkrét, véges állapotú \mathcal{X} Markov-lánc irreducibilis és aperiodikus, akkor \mathcal{X} stabil, és létezik egyetlen invariáns eloszlása, amely \mathcal{X} határérték-eloszlása (vagyis $p^{(\infty)}$ az egyetlen, nemnegatív megoldása $p^{(\infty)} = p^{(\infty)} P$ -nek és $p^{(\infty)}$ eloszlás).

A stabil Markov-láncok esetén ezt a határérték-eloszlást (p^∞, p^{inv}) $\pi(X)$ -szel jelöljük.

1.4. tétel. Ha a diszkrét, véges állapotú \mathcal{X} Markov-lánc stabil és $\bar{f} = E_{\pi(X)}[f(X)] < \infty$, akkor $P(\lim_{N \rightarrow \infty} \hat{f}_N = \bar{f} = 1)$, ahol $\hat{f}_N = \frac{1}{N} \sum_{t=1}^N f(X_t)$

1.8. definíció (Ergodicitás). *A diszkrét, véges állapotú \mathcal{X} Markov-lánc geometrikusan ergodikus, ha létezik $0 \leq \lambda < 1$ és $V(\cdot) > 1$ függvény, hogy*

$$\forall i : \sum_j |p_{ij}^{(t)} - \pi_j| \leq V(i)\lambda^t \quad (1.36)$$

A legkisebb ilyen λ -t a konvergencia sebességének hívjuk.

1.5. tétel. *Legyen \mathcal{X} egy diszkrét, véges állapotú, geometrikusan ergodikus (vagyis stabil is) Markov-lánc, annak invariáns eloszlásából $\pi(X)$ indítva, f pedig egy valós értékű függvény, amelyre igaz, hogy $E_\pi[f(X)^{2+\varepsilon}] \leq \infty$ valamely $\varepsilon > 0$ -ra. Ekkor $\bar{f} = E_\pi[f(X)]$, és $\sigma^2 = \text{Var}_\pi(f(X))$ jelölések mellett $\hat{f}_N = \frac{1}{N} \sum_{t=1}^N f(X_t)$ -re:*

$$\tau^2 = \sigma^2 + 2 \sum_{k=1}^{\infty} E_\pi[(f(X_0) - \bar{f})(f(X_k) - \bar{f})] \quad (1.37)$$

létezik, nemnegatív és véges, valamint

$$\sqrt{N} \frac{\hat{f}_N - \bar{f}}{\tau} \xrightarrow{d} N(0, 1), \text{ ha } N \rightarrow \infty \quad (1.38)$$

1.6.3. A Metropolis–Hastings-algoritmus

Jelölje $\pi(X)$ a normalizálatlan, szigorúan pozitív ($\forall i : \pi_i = \pi(X = i) > 0$) céleloszlást az $S = 0, 1, \dots, K$ halmaz felett. Legyen Q , egy átmenet-valószínűségi mátrix ($Q\mathbf{1} = \mathbf{1}$), az ú.n. javasolt eloszlás, úgy hogy $q_{ij} > 0$ akkor és csak akkor, ha $q_{ji} > 0$. Definiáljuk a χ Markov-láncot a P állapotátmenet-mátrixszal a következőképpen:

$$\forall i \neq j : p_{ij} = q_{ij} \min\left(1, \frac{\pi_j q_{ji}}{\pi_i q_{ij}}\right), \quad (1.39)$$

$\frac{0}{0} = 0$ -t feltéve és $p_{ii} = 1 - \sum_{j \neq i} p_{ij}$ definícióval. Fontos látni, hogy csak a π céleloszlás elemeinek arányaira van szükségünk, ami jól illeszkedik a bayesi analízis során előforduló normalizálatlan eloszlások alkalmazásához.

A definiált Markov-lánc stacionárius eloszlása $\pi(X)$ lesz, ami könnyen belátható, látván, hogy a részletes egyensúly-feltétel teljesül. Az $i = j$ és a $q_{ij} = q_{ji} = 0$ esetek triviálisan teljesülnek. Ha $i \neq j$ és $q_{ij} > 0$, akkor feltéve, hogy $\pi_i q_{ij} \geq \pi_j q_{ji}$:

$$\pi_i p_{ij} = \pi_i \frac{\pi_j q_{ji}}{\pi_i q_{ij}} = \pi_j q_{ji} = \pi_j p_{ji} \quad (1.40)$$

Ha Q irreducibilis, akkor P is az lesz, és ez hasonlóképpen igaz az aperiodikusságra is. Következésképpen, ha találunk egy olyan Q javaslati eloszlást, amely irreducibilis és aperiodikus, akkor egy adott $\pi(X)$ céleloszláshoz a fenti konstrukció egy stabil és reverzibilis Markov-láncot definiál, amelynek (invariáns) határeloszlása $\pi(X)$ lesz.

A Metropolis–Hastings-algoritmus alkalmazása tehát az alábbi fő lépésekből áll:

0. (Konstruáljuk meg a poszterior eloszlás egy P^S közelítését; ezt az MCMC inicializálására és később esetlegesen ellenőrzésre használhatjuk.)
1. Konstruáljunk egy irreducibilis és aperiodikus Q javaslati eloszlást, specifikusan a doménhez.
2. Sorsoljunk egy x_0 kiinduló állapotot P^S -ből.
3. $t = 1, 2, \dots$ -ra:
Sorsoljunk egy x^* jelölt állapotot Q -ból, feltéve x_t -t.
Számítsuk ki az α elfogadási valószínűséget az x_t -ből x^* -ba történő lépésre:

$$\alpha(x_t, x^*) = \min\left(1, \frac{\pi_{x^*} q_{x^*, x_t}}{\pi_{x_t} q_{x_t, x^*}}\right) \quad (1.41)$$

Legyen $x_{t+1} = x^*$ $\alpha(x_t, x^*)$ valószínűséggel, különben legyen $x_{t+1} = x_t$.

4. Folytassuk ezt, amíg az a lánc nem konvergál és el nem éri az előírt konfidenciát.
5. (Értékeljük ki a konvergencia sebességét és javítsunk a hatékonyságon Q újratervezésével; ezután lépünk vissza 2-re.)
6. (Vessük össze a módszert P^S fontossági újramintavételezésén alapuló alap módszerrel; ezután lépünk vissza 1-re.)

A Metropolis–Hastings algoritmus aletei

A fentebb ismertetett Metropolis–Hastings algoritmus a legáltalánosabb MCMC eljárás; bizonyos paramétereinek különféle beállításával számos, a szakirodalomban számon tartott speciális esetét kapjuk. Ezek közül a következőket kell megemlíteni:

Metropolis-algoritmus. Akkor áll elő, ha a Q állapotátmeneti mátrix szimmetrikus.

Véletlen bolyongás Metropolis-algoritmus. Ha Q csak a jelenlegi és a javasolt állapot között értelmezett valamilyen távolságtól függ: $q(x^*|x_t) = q(|x^* - x_t|)$.

Független mintavételező. Ha Q nem függ a jelenlegi állapottól: $q(x^*|x_t) = q(x^*)$.

Gibbs-mintavételező. Ha Q olyan, hogy az állapotvektornak egyszerre mindig csak egy elemét változtatja meg, annak a többtől való feltételes eloszlása szerint, 1 elfogadási valószínűséggel.

1.6.4. Következtetés Bayes-hálókbán Gibbs-mintavételezéssel

A Gibbs-mintavételező a következő lépéseket követi.

1. Kisorsolunk egy konfigurációt, amely kompatibilis a bizonyítékokkal.
2. Rögzítjük az evidencia-változók értékét, majd a fennmaradó változókon periodikusan:
3. Az aktuális változó értékét újrásorsoljuk a többi változó (praktikusan az aktuális változó Markov-takarója), mint feltétel szerint.
4. Minden egyes, a fenti lépés szerint kapott konfiguráció egy-egy eleme lesz a teljes mintahalmaznak.

1.7. Függelék: A következtetés komplexitása Bayes-hálókbán

Egy Bayes-háló „mérete” Ahhoz, hogy beszélni tudjunk egy eljárás tár- és időigényéről, meg kell tudnunk határozni a bemenet (esetünkben az adott Bayes-háló) hosszát. Egy Bayes-háló leírásához annak struktúráját, illetve a csomópontokhoz tartozó feltételes eloszlásokat kell megadnunk. Ezt megtehetjük úgy, hogy minden csomópontra megadjuk annak szüleit, és a CPT bejegyzéseit. Mivel ez utóbbiak száma a család (a gyermek és szülei) tagjai értékkészlete Descartes-szorzatának számossága, vagyis a szülők számában exponenciális, a strukturális információ mértékét elhanyagolhatjuk és tekinthetjük a Bayes-hálót meghatározó paraméterek számát az összes CPT bejegyzés számának. (Az, hogy az eloszlás-kényszer miatt egy-egy CPT mérete valójában $|gyerek| - 1$ -gyel arányos szintén elhanyagolható, hiszen csak egy polinom szorzót jelent.)

Mint fentebb láttuk, a következtetés komplexitása egy naív, az összes lehetséges változó-konfigurációt felsoroló eljárás esetében arányos ezek számával, ami a változók számában exponenciális, vagyis valós alkalmazások esetén kivitelezhetetlen. Az alábbiakban bemutatjuk, hogy az ismert *3SAT* probléma visszavezethető a Bayes-hálókbán való következtetésre, amiből következik, hogy az valóban NP-nehéz. A következőkben néhány, az algoritmikus problémák nehézségével és nehézségük összehasonlításával kapcsolatos alapfogalmat tekintünk át.

1.9. definíció (P). $P = \cup_{k \geq 1} \text{TIME}(n^k)$, ahol $\text{TIME}(\cdot)$ a Turing-gépek által a bemenetnek a paraméterben megadott függvényével arányos időben felismerhető (megoldható) nyelvek (problémák) osztálya.

Vagyis P azon nyelvek (problémák) halmaza, amelyek a bemenet hosszának valamilyen polinomiális függvényével arányos időben felismerhető (eldönthető). Általánosan kijelenthető, hogy a P-be tartozó problémák „könnyen”, „hatékonyan” megoldhatók, kezelhetők.

1.10. definíció (NP). $NP = \cup_{k \geq 1} \text{NTIME}(n^k)$, ahol $\text{NTIME}(\cdot)$ a nemdeterminisztikus Turing-gépek által a bemenetnek a paraméterben megadott függvényével arányos időben felismerhető (megoldható) nyelvek (problémák) osztálya.

Az NP-beli problémák tehát olyanok, amelyek esetén egy megoldásról polinom időben eldönthető, hogy az valóban helyes-e, arról azonban, hogy ezt a megoldást hogyan és mennyi idő alatt lehet megtalálni, már nem állítunk semmit. Mint a következőkben látni fogjuk, számos „nehéz” NP-beli probléma létezik, ha pedig ezeket vissza tudjuk vezetni egy másik (az általunk éppen vizsgált) problémára, akkor ily módon beláthatjuk (vagy legalábbis elfogadhatjuk), hogy a mi adott problémánk is ugyanolyan nehéz. A visszavezetés fogalmának precízebb definíciója az alábbi.

1.11. definíció (Karp-redukció). *Az f leképezés az L_1 nyelv Karp-redukciója az L_2 nyelvre, ha $\forall x$ szóra $x \in L_1$ pontosan akkor teljesül, ha $f(x) \in L_2$, és f polinom időben számítható. Jelölése: $L_1 \prec L_2$.*

1.12. definíció (NP-teljesség). *Egy L nyelv NP-teljes, ha eleme NP-nek és bármely $L' \in NP$ nyelvhez létezik $L' \prec L$ Karp-redukció.*

1.13. definíció. *Egy L nyelv NP-nehéz, ha létezik hozzá egy L' NP-teljes nyelv, hogy $L' \prec L$.*

Belátható, hogy ha egy adott L_1 NP-teljes nyelvhez létezik $L_2 \in NP$ nyelv, amelyre $L_1 \prec L_2$, akkor L_2 is NP-teljes.

Vagyis összefoglalva: ha tudunk adni egy polinomidejű algoritmust, amely egy ismert NP-nehéz problémát a mi problémaosztályunk egy elemére vezet vissza, azzal bizonyíthatjuk, hogy a mi problémánk is NP-nehéz.

1.7.1. A 3SAT probléma visszavezetése a Bayes-hálóban való következtetésre

Konjunktív normál formának (conjunctive normal form – CNF) hívjuk az olyan logikai kifejezéseket, amelyek literálok diszjunkcióinak („vagy” kapcsolatainak) konjunkciójaként („és” kapcsolataként) állnak elő, azaz például $(x_1 \vee \bar{x}_2 \vee x_3) \wedge (x_2 \vee x_4) \wedge \dots$ alakúak. Egy n -CNF egy olyan CNF melynek egy-egy tagjában maximum n literál szerepel.

A 3-CNF-ek kielégíthetőségét vizsgáló 3SAT problémáról (amely tehát azt vizsgálja, hogy egy adott 3-CNF változói behelyettesíthetőek-e úgy, hogy a kifejezés igaz legyen) ismert, hogy NP-teljes. Ha tehát tudunk egy eljárást adni, amely minden 3SAT problémát visszavezet egy Bayes-hálóban való következtetésre, akkor azzal igazoljuk, hogy a Bayes-hálóban való következtetés NP-nehéz.

Vegyük a következő kifejezést:

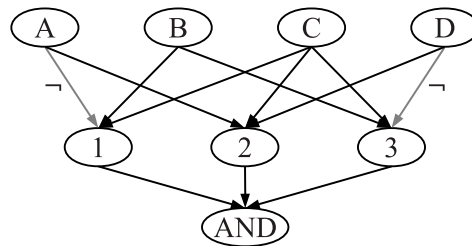
$$(\bar{A} \vee B \vee C) \wedge (A \vee C \vee D) \wedge (B \vee C \vee \bar{D}). \quad (1.42)$$

Ebből az 1.7 ábrán látható struktúrát konstruáljuk, vagyis:

- A háló minden csomópontja az {igaz, hamis} tartományból kaphat értéket.
- A kifejezésben szereplő változók a háló felső sorának csomópontjai lesznek, {0,5, 0,5} a priori eloszlással.

- Az egyes tagoknak egy-egy, a középső szinten elhelyezkedő csomópont felel meg, amely feltételes eloszlása az adott tag által leírt kifejezésnek lesz megfelelő (például az „1”-es csomópont 1 valószínűséggel *igaz* lesz, ha *B* és *C* közül legalább az egyik igaz, vagy ha *A* hamis, minden más esetben 1 valószínűséggel *hamis* lesz).
- A háló „AND” csomópontja az előző szinthez hasonlóan fogja össze a középső csomópontok értékét „és” kapcsolattal.

Belátható, hogy ebben a hálóban az $E = \{\text{AND} = \text{igaz}\}$ bizonyíték mellett azon behelyettesítéseknek lesz nem nulla a valószínűsége, amelyek a kifejezést igazzá teszik. Ebből már látható, hogy egy a leképezett 3SAT kifejezést kielégítő kifejezés keresése visszavezethető a konstruált hálóban való következtetésre. Ezzel beláttuk, hogy a következtetés általános esetben NP-nehéz.



1.7. ábra. A 3SAT probléma visszavezetése egy Bayes-háló struktúrára.

Irodalomjegyzék

- [1] Cecil Huang and Adnan Darwiche. Inference in belief networks: A procedural guide. *International Journal of Approximate Reasoning*, 15(3):225–263, 1996.
- [2] S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 50(2):157–224, 1988.