

Intelligens Rendszerek I. laboratórium (VIMIA360)

4. laborgyakorlat (Bayes-hálók)

BNet 101 SZOFTVERHASZNÁLATI ÚTMUTATÓ

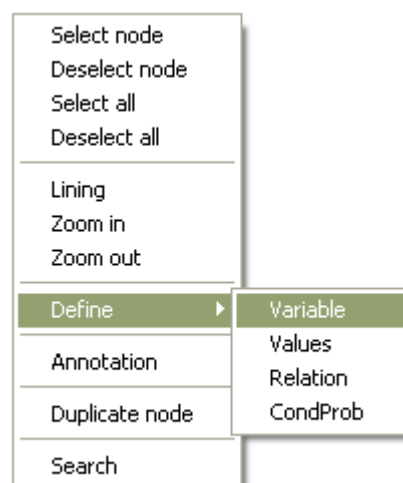
A következőkben röviden összefoglaljuk, hogy a kapcsolódó laborgyakorlat során egészen pontosan hogyan célszerű használni a gyakorlaton kapott **BNet** szoftvert annak érdekében, hogy minél kevesebb fejfájást okozzon. A laborgyakorlat résztvevői számára az alábbiak **BETARTÁSA KÖTELEZŐ**.

Szeretnénk előre is leszögezni, hogy a BNet egy tanszéki fejlesztésű, önálló kutatás-fejlesztési, kísérleti munka eredménye, nem pedig egy robusztus ipari alkalmazás (Antal Péternek (antal@mit.bme.hu) köszönhető, aki szabad- és munkaidejében, kezdetben elsősorban saját kutatásának megtámogatása érdekében fejlesztette ki ezt a világon egyedülálló, folyamatosan bővített/fejlesztett, és egyre szélesebb körben használt, egyre több Bayes-hálókkal kapcsolatos akár igen szofisztikált, legújabb kutatási trendekhez aktualizált funkcionalitást is támogató programot/programcsomagot). Tehát vegyük megtiszteltetésnek, hogy ezt az egyedülálló programot használhatjuk a laborgyakorlat keretében (arról nem is beszélve, hogy nincs is komolyabb alternatívája), semmint hogy bug-os, kiforratlan alkalmazásként tekintsünk rá. Minden funkció helyes működésre bírható ugyanis, ha az ember tudja, **hogyan** nyúljon az alkalmazáshoz (néha akár még az utasítások kiadásának sorrendje is számít). Epp ezért, a laborgyakorlaton résztvevő hallgatók életének megkönnyítése érdekében állítottuk össze ezt a rövid szoftverhasználati útmutatót, hogy a kapcsolódó laborgyakorlat során megvalósítandó Bayes-háló BNet-es létrehozását könnyebbé, gyorsabbá, és gördülékenyebbé tegyük. Vegyük tehát sorra, hogy mik is a teendőink (**szigorúan tartsuk be**)!

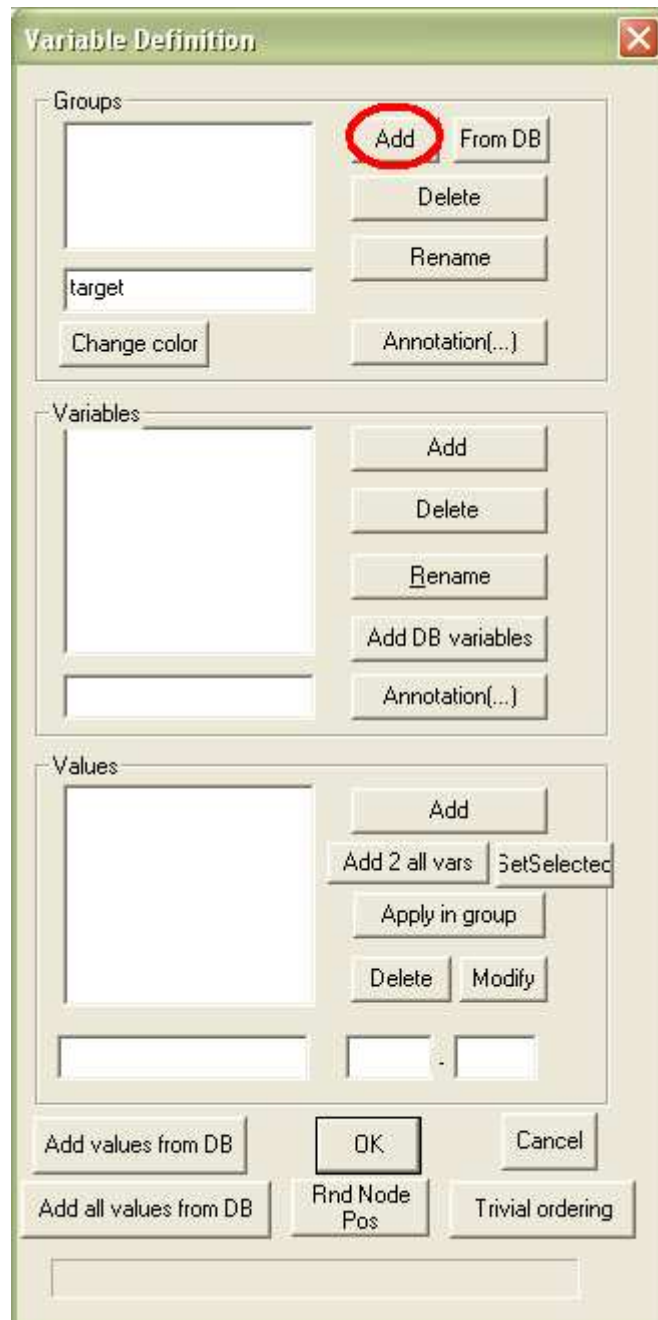
1. Először is a gyakorlat során egy bináris (azaz 2-elemű értékészlettel rendelkező valószínűségi változókat tartalmazó) Bayes-hálót kell terveznünk/rajzolnunk, majd pedig valószínűségi/bizonytalan/bizonytalanságot is figyelembe vevő következtetésre használunk. A hálóban a célváltozó (amivel kapcsolatban majd a főbb következtetéseket bonyolítjuk) adott és fix (flu), mindazonáltal a többi változót a feladatkiírásban szereplő készletből nekünk kell kiválasztani. Válasszuk tehát ki a hálóban szerepeltetni kívánt bináris valószínűségi változókat, és – Bayes-hálókkal kapcsolatos ismereteink, és tudásmérnöki, orvosi, páciensi és/vagy influenzával kapcsolatos eddigi tapasztalataink alapján – rajzoljunk fel ezek alapján **egyelőre még csak papíron** egy megfelelő, értelmes/ésszerű Bayes-hálót. Nagyon fontos, hogy a BNet-tel későbbiekben létrehozni kívánt Bayes-hálót teljes mértékben, végleges formában, előzetesen megtervezzük, hogy a BNet-tel már „csak” létre kelljen hozni, mondhatni implementálni. A BNet ugyanis nem a tervezést támogatja, hanem az implementációt. Tehát ne BNet-ben rajzolgassuk a hálót, hanem először tervezzük meg papíron, vagy bármilyen egyéb módon (pl. grafikus szoftverben, vagy akár fejben/gondolatban), és utána a BNet-ben már csak valósítsuk ezt meg. Mi kell ehhez?
 - a. Ki kell találnunk, hogy a flu változó mellett még milyen egyéb változók szerepeljenek a Bayes-hálóban;
 - b. Ki kell találnunk, hogy a változók között milyen oksági, valószínűségi, vagy akár kauzális kapcsolat van. Magyarán effektíve meg kell határoznunk a Bayes-hálónak megfelelő DAG (Directed Acyclic Graph) körmentes gráf éleit.

Meg kell adnunk, hogy melyik Bayes-háló beli csomópontnak melyik másik csomópont a szülője/gyermke. Köröket nyilván nem engedhetünk meg.

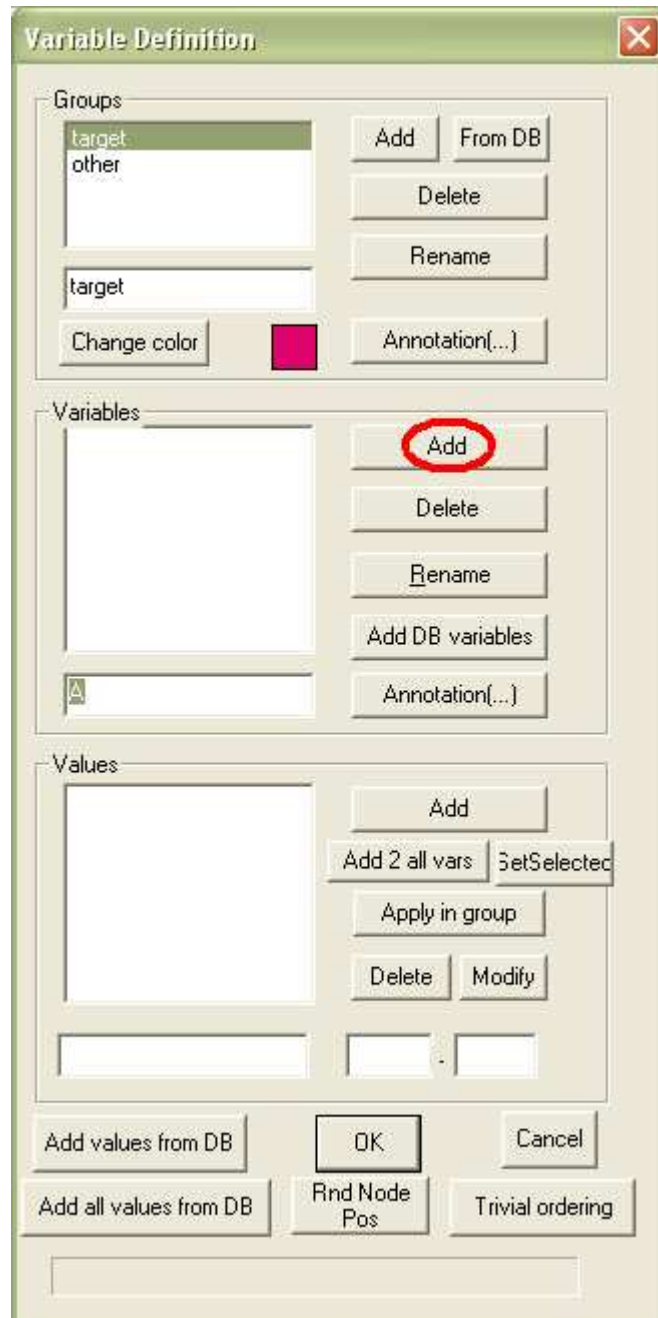
- c. A változók/csomópontok közti kapcsolatok megadását követően azt is specifikálnunk kell, hogy milyen valószínűségek szerepeljenek a Bayes-hálóban, pontosabban az egyes csúcsokhoz tartozó feltételes valószínűségi táblázatokban. Ugyebár egy-egy ilyen táblázat egy-egy változó kapcsán azt mondja meg, hogy egy-egy értékének fennállása mennyire valószínű a gyermek-változói egy-egy érték-konfigurációja (azaz a gyermekek egy-egy behelyettesítése) mellett. Ha tehát pl. egy bináris szülőnek 3 bináris gyermeke van, akkor összesen $2^3=8$ valószínűséget kell megadnunk, amelyek megmondják, hogy a 3 gyermek összes lehetséges érték-kombinációja esetén rendre mi a szülő értékeinek a valószínűsége (annak a valószínűsége, hogy amennyiben a gyermekek ezeket az értékeket vennék fel, úgy a szülő is az adott értéket venné fel). Természetesen itt ügyelnünk kell arra, hogy a gyermekek egy-egy érték-konfigurációja esetén a szülő változó értékei feletti feltételes valószínűségek teljes eloszlást adjanak (azaz a szülő változó értékeihez rendelt valószínűségek összege a gyermekek adott érték-konfigurációja mellett 1-nek adódjon). ...ezzel tehát el is készül a Bayes-hálónk terve/specifikációja, és máris nekifoghatunk az implementációnak.
2. Az előbbieken megtervezett Bayes-háló megvalósításához először is indítsuk el a BNet alkalmazást (C:\BNet\BNet.exe), majd zárjuk le az „stdout” nevű kis felugró ablakot.
3. Mivel a következőkben csak a BNet használatának munkamenetét szeretnénk szemléltetni, és nem mintamegoldást adunk a gyakorlathoz (nyilván), ezért a példa kedvéért most A, B, és C bináris változók használatára fogunk szorítkozni (ez az egyik legegyszerűbb, értelmes eset, amin érdemlegesen szemléltetni lehet a munkamenetet). ...hozzuk tehát első lépésben létre a változókat! Ehhez **kattintsunk ez egér jobb gombjával** a BNet-ben a fehér vászonra, és válasszuk ki a **Define → Variable** parancsot.



4. A felugró, „**Variables definition**” nevű ablakban, a „**Groups**” panelen belül írjuk be az első létrehozni kívánt változó-csoport nevét (példánkban ez target) a következő ábrán látható megfelelő mezőbe, és kattintsunk az „**Add**” gombra.



5. Az előbbiek eredményeképpen létrejön az első változó-csoportunk. Ezekhez a változó-csoportokhoz hozhatjuk majd létre a későbbiekben a változókat. Először is tehát minden változó-csoportot hozzunk létre, amit csak szeretnénk. Alapjában legalább kettőt javasunk: target (a célváltozókat magába foglaló csoport), és other (az egyéb változókat magába foglaló csoport). Hozzuk tehát az előbbihez hasonlóan létre most egymás után szépen sorban a többi változó-csoportot is! ... ügyeljünk az elnevezésekre (ne legyen bennük szóköz, ékezet, speciális karakter, kis betűvel kezdődjenek, stb)!
6. A változó-csoportok létrehozását követően hozzuk most létre a változókat! Kezdjük az elsővel: a „**Variables**” panelen belül írjuk be a létrehozni kívánt változó nevét (példánkban ez A) a következő ábrán látható megfelelő mezőbe miután előtte kijelöltük a „**Groups**” panelen belül a létrehozni kívánt változó csoportját, és kattintsunk az „**Add**” gombra.



7. Az előbbieket eredményeképpen létrejön az első változónk. Ezekhez a változó-csoportokhoz hozhatjuk majd létre a későbbiekben a változókat. Az előbbihez hasonlóan hozzuk most létre egymás után a többi változót is! ...és ügyeljünk az elnevezésekre (ne legyen bennük szóköz, ékezet, speciális karakter, lehetőleg kis betűvel kezdődjenek, stb)! Amennyiben esetleg nem jegyezte meg a program a változó-változócsoporthoz hozzárendeléseket (ezt a változókon végigkattintgatva ellenőrizhetjük), úgy sorban kattogtassuk őket végig, és mindegyiknél kattintsunk rá a változó csoportjára is. Példánkban tegyük fel, hogy az A változó csoportja a target (azaz célváltozó), míg a többi, B és C változók csoportja other (nem célváltozók).
8. Miután rendre létrejöttek a változók, a változócsoporthoz, és a köztük fennálló kapcsolat, külön mentsük le az eddig elért Bayes-hálót külön néven, hogy aztán

később az aggregatedDiagnoses változóval való kiegészítést innen tudjuk folytatni úgy, hogy visszatöltjük ezt a mentést, és innen nem túl nagy munkával újra felépítjük a hálónkat (javasolt egyébként minden egyes főbb lépés után létrehozni egy-egy újabb mentést, pl. alaphalo-01-groups.bn, alaphalo-02-vars.bn, extrahalo-02-vars.bn, extrahalo-03-vals.bn, stb).

- Most pedig hozzuk létre a változóink értékkészletét! Mivel bináris változókkal van dolgunk, ezért tegyük fel, hogy mindegyik értékkészlete azonosan: {false, true}. Mindegyik változóhoz sorban, fentről lefelé haladva létre kell hoznunk a változó értékkészletét. Ehhez kattintsunk mondjuk kezdetben a legfelső, először megadott változóra a **Variables** panelben, majd a **Values** panelben írjuk be első adandó értékének a nevét (esetünkben ez a false) az alábbi ábrán látható megfelelő mezőbe, majd kattintsunk az **Add** gombra.

The image shows a 'Variable Definition' dialog box with three main sections: Groups, Variables, and Values.

- Groups:** A list box contains 'target' and 'other'. To the right are buttons: 'Add', 'From DB', 'Delete', 'Rename', and 'Annotation[...]'. Below the list box is a text field containing 'target', a 'Change color' button with a pink color swatch, and another 'Annotation[...]'. button.
- Variables:** A list box contains 'A', 'B', and 'C'. To the right are buttons: 'Add', 'Delete', 'Rename', 'Add DB variables', and 'Annotation[...]'. Below the list box is a text field containing 'A'.
- Values:** A large empty list box. To the right are buttons: 'Add' (circled in red), 'Add 2 all vars', 'GetSelected', 'Apply in group', 'Delete', and 'Modify'. Below the list box is a text field containing 'false' and a separator field with a dot '.'.

At the bottom of the dialog are buttons: 'Add values from DB', 'Add all values from DB', 'OK', 'Cancel', 'Rnd Node Pos', and 'Trivial ordering'.

10. Hasonlóan hozzuk létre a változó másik, true értékét is, majd pedig a true értéket kijelölve írjuk be az érték megfelelő numerikus intervallumát is, ami most pontoszerűen 1 legyen, azaz az egy-elemű [1..1] zárt intervallum, amit az alábbi ábrán a megfelelő két beviteli mezőben „1.” és „1.” jelöl. Ezeket az 1-eseket tehát szépen írjuk be ebbe a két beviteli mezőbe a true érték kapcsán, és kattintsunk a **Modify** gombra.

The image shows a 'Variable Definition' dialog box with three main sections: Groups, Variables, and Values.

- Groups:** A list box contains 'target' and 'other'. To the right are buttons: 'Add', 'From DB', 'Delete', 'Rename', 'Change color' (with a pink color swatch), and 'Annotation(...)'. Below the list box is a text input field containing 'target'.
- Variables:** A list box contains 'A', 'B', and 'C'. To the right are buttons: 'Add', 'Delete', 'Rename', 'Add DB variables', and 'Annotation(...)'. Below the list box is a text input field containing 'A'.
- Values:** A list box contains 'false[0..0.]' and 'true[1..1.]'. To the right are buttons: 'Add', 'Add 2 all vars', 'SetSelected', 'Apply in group', 'Delete', and 'Modify' (circled in red). Below the list box is a text input field containing 'true'. Below this field are two numeric input fields, both containing '1.', which are also circled in red.

At the bottom of the dialog are buttons: 'Add values from DB', 'Add all values from DB', 'OK', 'Cancel', 'Rnd Node Pos', and 'Trivial ordering'.

11. Ezt követően sorban haladjunk végig a többi változón is (a **Variables** panelben), és ezeknek is adjuk meg az előbbiekhöz hasonlóan az értékkészletét. Ha ezzel készen vagyunk, kattintunk az **OK** gombra.

Variable Definition [X]

Groups

target
other
other

Change color [Blue]

Add From DB Delete Rename Annotation(...)

Variables

A
B
C
C

Add Delete Rename Add DB variables Annotation(...)

Values

false[0.;0.]
true[1.;1.]

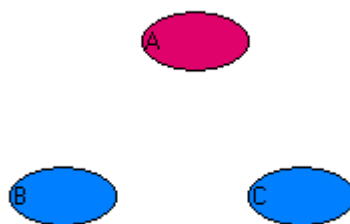
Add Add 2 all vars SetSelected Apply in group Delete Modify

Add values from DB Add all values from DB

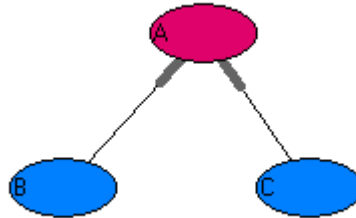
OK Cancel

Rnd Node Pos Trivial ordering

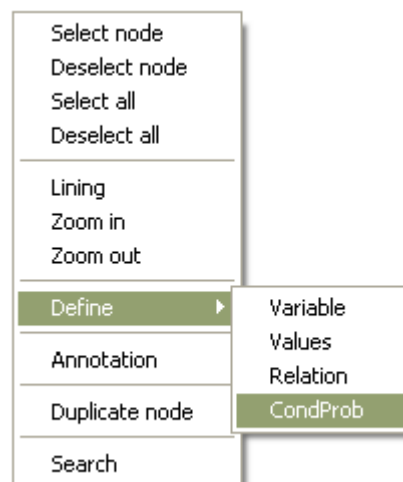
12. Megjelenik a vászon, és rajta a változóink. Tudva, hogy hogyan alakulnak köztük a kapcsolatok, helyezzük őket át a vásznon úgy, hogy lehetőleg minél inkább áttekinthető legyen a háló, az élek ne keresztezzék egymást, stb (**bal egérgombot** nyomva tartva egy-egy változón tudjuk **mozgatni**).



13. Miután szépen elrendeztük a változóinkat, húzzuk be köztük az éleket úgy, hogy azok valóban a kívánt szülő-gyermek kapcsolatot tükrözzék! A **Ctrl** gombot nyomva tartva, a **bal egérgombbal** a **SZÜLŐ**-változóból kiindulva húzzunk élet a **GYERMEK**-változóba. Végezzük ezt el minden élre!



14. Az előbbi ábrán tehát látható, hogy A közös gyermeke B-nek és C-nek, azaz értékeinek bekövetkezési valószínűsége (annak a valószínűsége, hogy az A változó a false vagy a true értéket veszi fel) feltételesen függ attól, hogy mi a B és C változó értéke, azaz pl.: $P(A=false|B=true, C=false) + P(A=true|B=true, C=false) = 1$. Ezeket a feltételes valószínűségeket feltételes valószínűségi táblázatokban (**Conditional Probability Table (CPT)**) minden egyes változó kapcsán meg kell adnunk. Azok a változók pedig, amelyeknek nincs szülője, nyilván apriori valószínűségeket kapnak az értékeikhez, nem pedig feltételeket. Ahhoz, hogy létrehozzuk ezeket a valószínűségeket, kattintsunk a vászonra az **egér jobb gombjával**, majd válasszuk ki a **Define → CondProb** parancsot.



15. A felugró „**Probability specification**” ablakban kattintsunk a „**Variable**” panelben lévő legördülő listára, és válasszuk ki belőle valamelyik szülő változót (esetünkben például az A változót).

Probability specification

Other variable:

Variable:

Condition (cc):

Rating:

Corr:

Point specification of conditional probabilities:

TypeScope:

Specification:

Confidence specification:

TypeScope:

Specification:

Description:

16. Itt „**Point specification and conditional probabilities**” panelben állítsuk be először is a **TypeScope**-ot **All** értékre, míg a tőle jobbra lévő legördülő lista értéke legyen **Table**.

Probability specification

Other variable: File

Variable:

Condition (configuration)

C	false[0	0:0.00	false[0..0.]
B	false[0	1:0.00	true[1..1.]

>> >>u >>e

Free Reorder

Rating:

Corr:

Point specification of conditional probabilities:

false[0..0.]	Table	0.0000	S:None
true[1..1.]	Table	0.0000	S:None

^ | u

f(x) mix

TypeScope:

Specification: Get similar

Confidence specification:

TypeScope: Set Weight

Specification: All-HP 1-HP

Description

Edit

Update implicites In Out Confidence OK Cancel

17. Ez az első, amit MINDEN VÁLTOZÓ KAPCSÁN meg kell tennünk, mielőtt még állítgatjuk az értékek valószínűségeit. **Állítsuk tehát be az előbbieket minden változóra!!**

18. Most pedig következhet a változók értékei valószínűségének megadása. Kezdjük a példa kedvéért az A változóval. Ez ugyebár gyermek (azaz vannak szülei, melyektől feltételesen függ), ezért a „**Condition (configuration)**” részben annyi sor szerepel, ahány szülője van. Esetünkben ez két sor, és mindkét sorban alapértelmezésben false érték szerepel. Ha kijelöljük bármelyik sort, akkor az ettől jobbra lévő kis részben szereplő értékek közül bármelyiket hozzárendelhetjük ideiglenesen az adott változóhoz. Ily módon beállíthatunk egy érték-konfigurációt (amely alapértelmezésben minden változóhoz a false értéket társítja). Haladjunk tehát végig a „**Variable**” részben aktuálisan kiválasztott változó szüleihez tartozó minden érték-konfiguráción, és ezeknél rendre adjuk meg a gyermek mindegyik értékének a valószínűségét (ügyelve arra, hogy a gyermek értékei felett teljes valószínűség eloszlás álljon elő). A példa kedvéért tegyük fel, hogy az A változó false értékéhez a B és C változók false-false konfigurációja mellett a 0.3 valószínűséget rendeljük,

aminek következtében a teljes eloszlásból adódik, hogy az A változó true értékéhez viszont 0.7 valószínűség kellene, hogy tartozzon az adott szülő-konfiguráció mellett (ezt mi is beállíthatjuk manuálisan, de a program is beállítja automatikusan, ha miután bevittük az A változó false értékéhez tartozó **0.3** valószínűséget, és nyomtunk egy **ENTER**-t, és így már az A változó true értéke van kijelölve, rákattintunk a kis „!” jelre).

Probability specification

Other variable: File

Variable:

Condition (configuration)

C	false[0. 0.00	false[0. 0.00	>> >>u >>e
B	false[0. 1.00	true[1. 1.00	Free Reorder

Rating:
Corr:

Point specification of conditional probabilities:

false[0. 0.00	Table	0.3000	S:None	^ ! u
true[1. 1.00	Table	0.7000	S:None	f(x) mix

TypeScope:

Specification: Get similar

Confidence specification:

TypeScope: Set Weight

Specification: All-HP 1-HP

Description: Edit

Update implicites In Out Confidence OK Cancel

19. A „**Condition (configuration)**” panelben található >> gombra kattintva léptessük tovább a szülő változók érték-konfigurációját, és arra is adjuk meg a gyermek-változó értékei feletti teljes eloszlást. **Haladjunk végig ennek megfelelően a gyermek-változó összes szülő-konfigurációján!**
20. Miután elkészültünk egy változóval (azaz például egy gyermek-változó esetén minden szülő-konfigurációja esetén megadtuk a lehetséges értékeinek az eloszlását), akkor lépünk tovább a **Variable** panelben a következő változóra, és annak is hasonlóképpen adjuk meg az összes valószínűségét. Olyan változók esetén, melyeknek nincs szülője, nyilván nem kell „pörgetnünk” a szülők érték-konfigurációját, azaz az előbbiekhöz

hasonlóan apriori valószínűségeket rendelhetünk az értékeikhez (nyilván ennek is teljes eloszlásnak kell megfelelnie). Ha elkészültünk minden változóval, azaz mindegyik változó mindegyik értékének minden esetben adott a bekövetkezési valószínűsége, úgy kattintsunk az **OK** gombra.

Probability specification

Other variable: File

Variable:

Condition (configuration)

>> >>u >>e

Free Reorder

Rating: 0

Corr: 0

Point specification of conditional probabilities:

false[0..0.]	Table	0.5000	S:None
true[1..1.]	Table	0.5000	S:None

^ ! U

f(x) mix

TypeScope: All Table

Specification: 0.5000 Get similar

Confidence specification:

TypeScope: Local None Set Weight

Specification: All-HP 1-HP

Description Edit

Update implicites In Out Confidence **OK** Cancel

21. Ezzel elkészült a Bayes-hálónk, akár le is menthetjük, vagy végre is hajthatunk benne egy következtetést. Ehhez az **Inference** menü **Inference** parancsára kell kattintani, majd a felugró „**Expression specification**” ablakban meg kell adnunk az egyes megfigyeléseinket, avagy az **evidenciákat** (a megfelelő változókat kijelölve a **Variables** részben, illetve kijelölve megfelelő/megfigyelt értéküket a **Values** részben, és a **Hard** gombra kattintva), amiknek fényében ki szeretnénk következtetni a többi változó értékeinek valószínűségét. Ehhez a **Values** részben az **Update** gombra kell kattintanunk, majd sorra végiglépkedhetünk a változókon és megtekinthetjük, hogy mi értékeik valószínűsége a megadott esetben. Ha pedig egy evidenciát vissza szeretnénk venni, akkor a Hard gombtól balra lévő **Uni.** gombra kell kattintanunk.

Expression specification

Variables:

A	-	MP:
B	false[0.:0.]	MP:
C	false[0.:0.]	MP:false[0.:0.]

Max Probability Set

☒ Variable in MP

CP(MP config):

Update

Values:

false[0.:0.]	E:1.0000 P:1.0000 T:N
true[1.:1.]	E:0.0000 P:0.0000 T:N

Confidence

Update

Collective Probability Set

☐ Value in CPS

CP(Collective)

Update

Virtual evidence

Norm.

Unj.

Hard

Set sample

1.0000

Method:

Tree of Cliques (exact,default)

Close

Expression specification ✖

Variables:

A	-	MP:Unknown
B	false[0.;0.]	MP:Unknown
C	false[0.;0.]	MP:Unknown

Max Probability Set

☒ Variable in MP CP(MP config): Update

Values:

false[0.;0.]	E:1.0000	P:0.3000	C:N
true[1.;1.]	E:1.0000	P:0.7000	C:N

Confidence Update

Collective Probability Set

☐ Value in CPS CP(Collective) Update

Virtual evidence

Norm. Unj. Hard Set sample

Method: Tree of Cliques (exact,default) Close

Készítette:
Kovács Dániel László
dkovacs@mit.bme.hu
Budapest, 2011.03.01