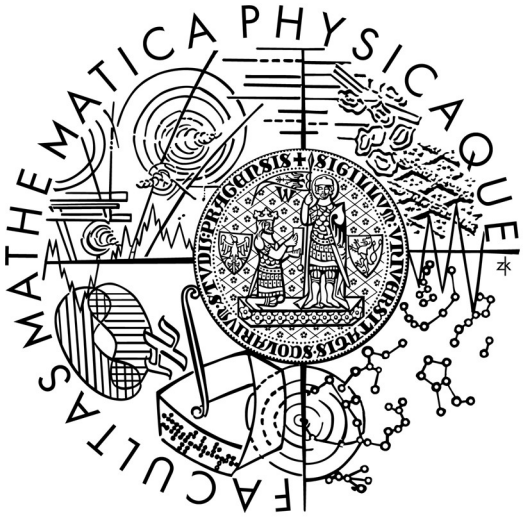
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

DIPLOMOVÁ PRÁCE



Peter Lacký

Teoretické způsoby modelování uživatelského rozhodování

Katedra softwarového inženýrství

Vedoucí diplomové práce: Prof. RNDr. Peter Vojtáš, DrSc.

Studijní program: Informatika

Na tomto mieste by som rád poďakoval vedúcemu diplomovej práce Prof. RNDr. Petrovi Vojtášovi, DrSc. a konzultantovi Mgr. Alanovi Eckhardtovi za ich rady a pripomienky, ktoré mi pomohli pri vytváraní tejto práce. Ďalej ďakujem rodičom, že ma podporovali počas celej doby štúdia, a všetkým priateľom, s ktorými som túto prácu konzultoval.

Prehlasujem, že som svoju diplomovú prácu napísal samostatne a výhradne s použitím citovaných prameňov. Súhlasím so zapožičiavaním práce.

V Prahe dňa 4. decembra 2008 Peter Lacký

Obsah

Název práce: Teoretické způsoby modelování uživatelského rozhodování

Autor: Peter Lacký

Katedra (ústav): Katedra softwarového inženýrství

Vedoucí diplomové práce: Prof. RNDr. Peter Vojtáš, DrSc.

e-mail vedoucího: Peter.Vojtas@mff.cuni.cz

Abstrakt: Táto práca sa zaoberá problematikou modelovania užívateľských preferencií. Obsahuje rozbor rozdielnych pohľadov na užívateľské preferencie. Práca obsahuje prehľad stávajúcich modelov užívateľských preferencií a porovnania medzi nimi. Podrobne rozoberá Fuzzy Logické Programovanie, Bayesove Logické Programovanie, Pravdepodobnostné Relačné Modely a Markovove Logické Siete. Pre jednotlivé modely sú navrhnuté transformácie do iných modelov a taktiež sú ukázané ich možnosti použitia v reálnom svete. V závere práce sú uvedené návrhy na rozšírenia jednotlivých modelov.

Klíčová slova: užívateľské preferencie, fuzzy a pravdepodobnostné modely užívateľských preferencií, pravdepodobnostné rozhodovanie, Bayesove a Markovove siete

Title: Theoretical aspect of modelling of user decision

Author: Peter Lacký

Department: Department of Software Engineering

Supervisor: Prof. RNDr. Peter Vojtáš, DrSc.

Supervisor's e-mail address: Peter.Vojtas@mff.cuni.cz

Abstract: In this thesis we address to the problematics of modelling user preferences. We discuss different views on user preferences as well as we give an overview of known models of user preferences and compare them. In more detail we introduce Fuzzy Logic Programming, Bayesian Logic Programming, Probabilistic Relational Models and Markov Logic Networks. For each model we propose transformations to other models and we show possible utilizations in real world. Finally we present our suggestions how to extend and improve these models.

Keywords: user preferences, fuzzy and probabilistic models of users preferences, probabilistic reasoning, Bayesian and Markov network

1 Úvod

V dnešnej dobe má každý užívateľ internetu prístup k obrovskému množstvu informácií a možnostiam na uľahčenie každodennej práce. Keďže v dnešnej dobe internetu počet týchto možností rýchlo narastá, je čím ďalej náročnejšie nájsť medzi nimi práve tie, ktoré konkrétnemu užívateľovi najviac vyhovujú na základe jeho preferencií.

Spôsoby ako reprezentovať užívateľské preferencie sa nazývajú modely. Úlohou týchto modelov je, okrem definovania reprezentácie užívateľských preferencií, poskytnúť možnosť istých predpovedí. Takže konkrétny model obsahujúci už nejaké informácie (vlastnosti, preferencie a vzťahy medzi nimi) o užívateľovi alebo o skupinách užívateľoch, poskytuje odpovede na otázky čo je najvhodnejšie (respektíve pravdepodobnostne najvhodnejšie) pre užívateľa.

Touto problematikou sa zaoberá oblasť skúmania s názvom užívateľské preferencie, ktorú je možné rozdeliť na dve časti.

- ako reprezentovať jednotlivé modely s použitím známych techník, za účelom zvyšovania efektivity a presnosti predpovedí
- ako zisťovať závislosti medzi jednotlivými preferenciami a následne ich spätná adaptácia do konkrétneho model

Zisťovanie závislosti a ich spracovanie sú tvz. induktívne metódy. Hromadne sa táto oblasť skúmania s názvom učenie (Learning).

Cieľom je zostaviť prehľad, akými spôsobmi je možné reprezentovať užívateľské preferencie. Keďže táto úloha je veľmi rozsiahla, boli vybrané len niektoré modely vychádzajúce z rozdielnych teórií. Následne sú tieto modely porovnané a navzájom konfrontované na rovnakom príklade. Z dôvodu rozsiahlosti problematiky modelovania užívateľských preferencií sa táto práca nezaoberá induktívnymi metódami. Táto problematika učenia jednotlivých modelov už presahuje rozsah zadania tejto diplomovej práce. Týmto problémom sa zaoberá napr. Práca [4].

Táto diplomová práca patrí do kategórie porovnávacích prác a vychádza z čisto teoretického základu a to v oblasti, v ktorej doposiaľ nebola podobná prehľadová a porovnávacia práca spracovaná. Preto neobsahuje žiadnu implementáciu modelov.

V kapitole 2 sú priblížené základné pojmi potrebné k všeobecnej interpretácii

jednotlivých modelov, od výrokovej logiky cez predikátovú logiku až k logickému programovaniu a viachodnotovej (konkrétne fuzzy) logike. Ďalej základné pojmi z teórie pravdepodobnosti a nakoniec popis Bayesových a Markovových sietí.

Kapitola 3 obsahuje pohľad na preferencie obecne a možnosti ich využitia v reálnom svete. Taktiež približuje myšlienku sémantického webu, ako nástroja na sprehľadnenie internetu a zefektívnenie jeho interakcie s užívateľom.

V kapitole 4 je zostavený stručný prehľad jednotlivých modelov, taktiež táto kapitola obsahuje prehľad už dokázaných ekvivalencií medzi jednotlivými modelmi. Zo všetkých modelov sú bližšie rozobrané len niektoré a to také, ktoré poskytujú silnejšie nástroje ako ostatné, pokrývajú najširšiu skupinu modelov a sú dobre definované.

Kapitola 5 obsahuje porovnanie medzi jednotlivými modelmi, ktoré sú bližšie predstavené v kapitole 4. Taktiež obsahuje možnosti ako transformovať jednotlivé modely medzi sebou. A nakoniec stručné hodnotí výhody a nevýhody jednotlivých modelov.

V kapitole 6 je ukázaný príklad použitia jednotlivých modelov v praxi (v reálnom živote) a to na príklade zákazníka (turistu), ktorý si chce pre seba vybrať najvhodnejší zájazd. V záverečnej kapitole 7 sú postrehy, ako rozšíriť jednotlivé modely, a taktiež návrhy, ako vytvoriť obecnejšie a silnejšie modely.

2 Základné pojmy

K formálnemu popisu jednotlivých modelov užívateľských preferencií je potrebné formálne zadefinovať teóriu predikátovej logiky, z ktorej vychádza logické programovanie skúmajúce logické odvodzovanie. V tejto kapitole sú ďalej priblížené pojmi z teórie pravdepodobnosti keďže samotné modely predstavujú istú neistotu a pravdepodobnosť rozhodnutia. Nakoniec sú formálne zadefinované Bayesove a Markovove siete, ktorých teória je využitá v niektorých modeloch.

2.1 Výroková a Predikátová logika

Výrok je tvrdenie, alebo jazykový výraz, o ktorého pravdivosti (alebo nepravdivosti) má zmysel uvažovať. *Výroková logika* sa zaoberá, spôsobmi tvorenia výrokov pomocou spojok a vzťahmi medzi pravdivosťou rôznych výrokov. K tomu používa špecifický *jazyk výrokovej logiky*. Prvotná formula P je neprázdna množina, ktorej prvky môžu byť slová nejakého formálneho jazyka alebo len písmená p, r, q, p_1, p_2, p_3, . . . Prvky množiny P budú nazývané *prvotné formule* (tiež *výrokové premenné* alebo *elementárne výroky*). *Jazyk L_P výrokovej logiky nad množinou P* (inak *P je množina prvotných formúl jazyka L_P*) obsahuje:

- prvky množiny P
- *symboly pre logické spojky* ¬ *negácia,* & *konjunkcia,* ∨ *disjunkcia*, → *implikácia,* ↔ *ekvivalencia*
- *pomocné symboly (zátvorky)*

Samotná syntax jazyka L_P nad množinou prvotných formúl P indukciou definuje nasledujúce typy formúl

- *prvotné formule*
- *literály* sú prvotné formule a negácie prvotných formúl
- *klauzule* sú disjunkcie literálov

ďalej tiež *formula je v konjunktívnom tvare*, ak je to konjunkcia klauzúl a *formula je v disjunktívnom tvare*, ak je to disjunkcia konjunkcii literálov. Ďalej *Hornova klauzula* je klauzula s najviac jedným pozitívnym literálom a v tvare

$$H \vee \neg B_1 \vee, \ldots, \vee \neg B_n \quad \text{alebo} \quad (B_1, \ldots, B_2) \rightarrow H \quad \text{v skratke} \quad \boldsymbol{B} \rightarrow H$$

atóm H sa nazýva *hlava* (*head*) a n-tica atómov $B_1, \ldots, B_n$ sa nazýva *telo* (*body*) klauzule.

Typy Hornových klauzúl:

- *Fakt* – s pozitívnym a bez negatívnych literálov (tiež *goal clause*)
- *Pravidlo* – s pozitívnym a aspoň jedným negatívnym literálom
- *Cieľ* alebo *Dotaz* – bez pozitívneho literálov

Pravidlá alebo fakty s usporiadanou postupnosťou literálov sú nazývané *usporiadané klauzule* (tiež *definite clause*). Pre potreby strojového dokazovania splniteľnosti formúl bola navrhnutá efektívnejšia metóda ako použitím axiómou a pravidla modus ponens a to tak zvaná *rezolučný princíp* obsahujúci jediné pravidlo, ktoré je intuitívne efektívnejšie ako dokazovanie pomocou axiómou. Logické programovanie sa obmedzuje na klauzule konkrétneho typu a to na *Hornove klauzule* používané napríklad v programovacom jazyku Prolog.

Definícia (Lineárna vstupná rezolúcia): Nech P je množina Hornových klauzúl faktov alebo pravidiel a G cieľová klauzula. *Lineárna vstupná rezolúcia* (*LI-rezolúcia*) množiny $S = P \cup \{G\}$ je lineárne vyvrátenie S, ktoré začína klauzulou G a bočnými klauzulami sú len klauzule z P.

Definícia (LD-rezolučné vyvrátenie): Ak je $P \cup \{G\}$ množina usporiadaných klauzúl, potom *LD-rezolučné vyvrátenie* $P \cup \{G\}$ je postupnosť $<C_0, B_0>, \ldots, <C_n, B_n>$ taká, že $G_0 = G$, G_{n+1} = koreňovej klauzuly $C_i \in P$ a každé G_i , $1 \leq i \leq n$ je rezolventou usporiadaných klauzúl G_{i-1} , C_{i-1} .

Problém výberu vhodného literálu podľa ktorého sa bude rezolvovať je riešený *selekčným pravidlom R*, ktoré predstavuje ľubovolnú funkciu, ktorá vyberie literál z usporiadanej cieľovej klauzule. Potom *SLD-rezolúcia*je je lineárna vstupná rezolúcia so selekčným pravidlom.

Definícia (SLD-rezolučné vyvrátenie[1]): *SLD-rezolučné vyvrátenie* $P \cup \{G\}$ pomocou selekčného pravidla R je LD-rezolučné vyvrátenie $<C_0, B_0>, \ldots, <C_n, B_n>$, $G_0 = G$, G_{n+1} = koreňovej klauzuly, kde $R(G_i)$ je literál na ktorom rezolvujeme v i + 1 kroku. Bez explicitnej definície R je vybraný literál najviac vľavo.

Predikátová logika je založená na jazyku 1. rádu, ktorý vznikne rozšírením jazyka výrokovej logiky o predikátové symboly a kvantifikátory (existenčný a univerzálny).

1 Selected Linear Resolution for Definite Clause

Definícia (Jazyk 1. rádu): Jazyk 1. rádu obsahuje:

- *premenné* $x, y, z, x_1, y_1, z_1, \ldots$ ktorých je neobmedzene
- *funkčné symboly* $f, g, h, \ldots$ pre každý funkčný symbol je dané prirodzené číslo $n \geq 0$ (*arita funkčného symbolu*), ktoré vyjadrujú jeho početnosť
- *predikátové symboly* $p, q, r, \ldots$ pre každý predikátový symbol je dané prirodzené číslo $n \geq 1$ (*arita predikátového symbolu*), ktoré vyjadrujú jeho početnosť
- *logické spojky* $\neg$ *negácia,* $\&$ *konjunkcia,* $\vee$ *disjunkcia*, $\rightarrow$ *implikácia,* $\leftrightarrow$ *ekvivalencia*
- *kvantifikátory* $\forall$ *univerzálny,* $\exists$ *existenčný*
- *pomocné symboly* (,) , [,] , { , } , . . .

Funkčný (predikátový) symbol arity n je nazývaný *n-árny funkčný (predikátový) symbol.* Ľubovolná konečna postupnosť symbolov bude nazývaná v skratke *slovo* alebo *výraz. Termom* sa rozumie induktívna konštrukcia pomocou pravidiel:

i. Každá premenná je term.

ii. Ak sú výrazy $t_1, \ldots, t_n$ termy a f n-árny funkčný symbol, potom výraz $f(t_1, \ldots, t_n)$ je term.

iii. Každý term vznikne konečným použitím pravidiel (i) a (ii).

Term obsahujúci žiadne premenné sa nazýva *základný* (*ground*) *term.* Konštrukcia *formule* je definovaná podobne indukciou pomocou pravidiel:

i. Ak je p n-árny predikátový symbol a ak sú výrazy $t_1, \ldots, t_n$ termy, potom výraz $p(t_1, \ldots, t_n)$ je *atomická formula.*

ii. Ak sú výrazy A a B formule, potom výrazy $(\neg A)$, $(A \,\&\, B)$, $(A \vee B)$, $(A \rightarrow B)$, $(A \leftrightarrow B)$ sú tiež formule.

iii. Ak je x premenná a A je formula, potom $(\forall x)\, A$ a $(\exists x)\, A$ sú formule.

iv. Každá formula vznikne konečným použitím pravidiel (i) – (iii).

Definícia (Interpretácia jazyka): *Interpretácia I* nad jazykom 1. rádu L obsahuje

- neprázdnu množinu D, nazývanú *doménou (*tiež *univerzum)* interpretácie
- zobrazenie ktoré, každej konštante $c \in L$ priradí prvok z D

- zobrazenie $f_I : D^n \rightarrow D$ pre každý n-árny funkčný symbol jazyka L
- n-árnu reláciu $p_I \subseteq D^n$ pre každý n-árny predikátový symbol p jazyka L, okrem symbolu pre rovnosť

Prvky množiny D sú nazývané *individuá.* Zobrazenie e_I, množiny všetkých premenných do množiny D (*domény*) pri interpretácii I bude nazývané *ohodnotenie premenných pri interpretácii I.* Ak je e ohodnotenie premenných, x premenná a $m \in D$, potom ohodnotenie premenných, ktoré premennej x priraďuje indivíduum m a na všetkých ostatných ohodnoteniach splýva s ohodnotením e bude značené $e(x/m)$. Ohodnotenie e priraďuje hodnotu všetkým premenným (značenie $e(x)$ pre premennú x) Ohodnotenie e priraďuje termu t hodnotu $t[e]$.

Definícia (Interpretácia termov): Nech I je interpretácia, e ohodnotenie a t term.

- ak je t konštanta c, potom $t[e_I] = c_I$
- ak je t premenná x, potom $t[e_I] = e_I(t)$
- ak je t tvaru $f(t_1, \ldots, t_n)$ a hodnoty $t_1[e_I], \ldots, t_n[e_I]$ sú známe, potom $t[e_I] = f_I(t_1[e_I], \ldots, t_n[e_I])$

Ak je A formula a x premenná, potom výraz $(\exists x) B$ je skratka pre výraz $\neg(\forall x) \neg A$. Logické spojky sú redukované rovnako ako vo výrokovej logike na spojky ¬ a → alebo pre potreby Hornových klauzúl na $\vee$ a ¬ .

Definícia (Tarski – pravdivosť a splniteľnosť formúl): Nech I je interpretácia, e ohodnotenie a A formula.

i. Indukciou podľa zložitosti formule A bude definovaná *pravdivosť formule A v interpretácii I pri ohodnotení e.* Značenie $I \models A[e]$.

   a) Ak je A atomická formula tvaru $p(t_1, \ldots, t_n)$, kde p je n-árny predikátový symbol, rôzny od symbolu pre rovnosť a $t_1, \ldots, t_n$ sú termy, potom $I \models A[e]$, ak $(t_1[e], \ldots, t_n[e]) \in p_I$.

   b) Ak je A atomická formula tvaru $t_1 = t_2$ potom $I \models A[e]$, ak $t_1[e] = t_2[e]$, kde obe termy sú interpretované rovnakým indivíduum.

   c) Ak je A tvaru $\neg B$, potom $I \models A[e]$, ak $I \not\models B[e]$.

   d) Ak je A tvaru $(B \rightarrow C)$, potom $I \models A[e]$, ak $I \not\models B[e]$ alebo $I \models C[e]$.

e) Ak je A tvaru $(\forall x)B$, potom $I \models A[e]$, ak pre každé indivíduum $m \in D$ je $I \models B[e(x/m)]$.

ii. Formula *A je splnená v I* (zápis $I \models A$), ak je A pravdivá v I pri ľubovolnom ohodnotení *e.*

Pravdivosť formule záleží len na ohodnotení voľných premenných, ktoré sa v nej vyskytujú. Pravdivosť uzavretej formule nezávisí na ohodnotení vôbec. Formula *A je tautológia* (*logicky pravdivá*), ak je A pravdivá v pri každej interpretácii jazyka.

Definícia (Model interpretácie): Nech P je množina uzavretých formúl. Interpretácia I je nazývaná *modelom P,* ak každá formula z P je pravdivá v interpretácii *I.*

Jedným zo spôsobov určenia splniteľnosti množiny formúl vychádza z axiómou a odvodzovacích pravidiel a to Modus ponens a pravidla generalizácie. Ľahší spôsob k určeniu splniteľnosti množiny formúl je založený na Hebrandovej vete využívajúcej Herbrandov model a interpretáciu.

Definícia (Herbrandovo univerzum, Herbrandova báza): Nech L je jazyk prvého rádu. *Herbrandove univerzum* U_L *pre* L je množina všetkých termov bez premenných, ktoré je možné vytvoriť z konštánt a funkčných symbolov jazyka L. Množina B_L všetkých atomických formúl bez premenných je nazývaná *Herbrandova báza* B_L *pre* L.

Definícia (Herbrandova interpretácia): *Herbrandova interpretácia* je ľubovolná interpretácia I, ktorá priraďuje

- premenným prvky U_L
- konštantám priraďuje samé seba
- funkčným symbolom priraďuje funkciu $f_I(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$
- predikátovým symbolom ľubovolnú funkciu z U_L do pravdivostných hodnôt

Definícia (Herbrandov model): Nech L je jazyk prvého rádu a S množina uzavretých formúl. *Herbrandov model pre S* je Herbrandova interpretácia pre L, ktorá je modelom pre S. Inak tiež je to taká Herbrandova interpretácia, že každá formula z S je v nej pravdivá.

Veta (Herbrandova): Nech S je množina uzavretých formúl, potom buď, existuje Herbrandov model alebo existuje konečná množina formúl, tvorená prvkami S, ktorých konjunkcia neplati.

K rozhodnutiu o splniteľnosti množiny formúl, už nie je potrebné brať v úvahu všetky možné interpretácie, stačí pracovať len so symbolickými Herbrandovými interpretáciami.

Definícia (Substitúcia): *Substitúcia* θ je konečná množina dvojíc tvaru $\{x_1/t_1, \ldots, x_n/t_n\}$, kde $x_1, \ldots, x_n$ sú premenné a $t_1, \ldots, t_n$ sú termy a platí $x_i \neq t_i$ a $x_i \neq x_j$ ak $i \neq j$.

Množina premenných $\{x_1, \ldots, x_n\}$ je nazývaný *definičným oborom* θ (značenie *Dom*(θ)) a množina termov $\{t_1, \ldots, t_n\}$ je nazývaná *oborom hodnôt* θ (značenie *Range*(θ)).

Definícia (Kompozícia substitúcie): Nech $\theta = \{u_1/s_1, \ldots, u_n/s_n\}$ a $\psi = \{v_1/t_1, \ldots, v_m/t_m\}$ sú substitúcie. Potom kompozícia substitúcií (tiež zložená substitúcia) je množina $\theta\psi = \{u_1/s_1\psi, \ldots, u_n/s_n\psi, v_1/t_1, \ldots, v_m/t_m\}$, bez všetkých $u_i/s_i\psi$, pre ktoré $u_i = s_i\psi$ a bez všetkých v_j/t_j pre ktoré $v_j \in \{u_1, \ldots, u_n\}$. Pre ľubovolný výraz E a substitúcie θ, ψ a σ platí $(E\theta)\psi = E(\theta\psi)$ a $(\theta\psi)\sigma = \theta(\psi\sigma)$.

2.2 Logické programovanie

Logické programovanie je teória, ktorá podrobne skúma logické odvodzovanie (napríklad v Prologu, ale obecnejšie). V logickom programovaní sú používané čiarky (,) namiesto symbolu (&) pre konjunkciu a obrátená šípka (←) namiesto implikácie.

Definícia (Logický program): *Logický program* je ľubovolná konečná množina Hornových klauzúl. (Prologovský) program *P* je zoznam programových klauzúl faktov alebo pravidiel kde:

- *Fakt*: deklaruje vždy pravdivé veci

  clovek(peter, 24, student).

- *Pravidlo*: deklaruje veci, ktorých pravdivosť závisí na daných podmienkach

  studuje(X) :- clovek(X, _Vek, student).

- *Dotaz* (cieľ): užívateľ sa pýta programu, či sú veci pravdivé

  ?- studuje(peter). % yes splniteľný dotaz

  ?- studuje(lacky). % no nesplniteľný dotaz

Odpoveď na dotaz môže byť:

- pozitívna – dotaz je splniteľný a uspel
- negatívna – dotaz je nesplniteľný a neuspel

2.3 Fuzzy logika

Viachodnotová logika vznikne rozšírením predikátovej logiky, z dvojhodnotovej pravdivostnej množiny hodnôt {0, 1} na viachonotovú množinu. Fuzzy logika patrí medzi viachodnotové logiky, kde množinu pravdivostných hodnôt tvorí interval [0, 1]. Teóriu fuzzy logiky prvý krát definoval Lotfi A. Zadeh [18][17].

V rámci fuzzy logiky sa nehovorí o pravdivosti, alebo nepravdivosti, ale o *stupni pravdivosti* (*stupeň členstva*). Ako príkladom nech je 100 ml pohár, v ktorom je 30 ml vody. Potom z pohľadu toho, aký je pohár plný alebo prázdny má z pohľadu fuzzy logiky dve rozdielne hodnoty. Pohár je na 0,3 plný a na 0,7 prázdny. Z iného pohľadu plností pohára, aký je poloplný alebo poloprázdny, nadobúda hodnotu 1 pre 50 ml vody, takže pre 30 ml je pohár na 0,6 poloplný a poloprázdny.

Definícia (Fuzzy množina): *Fuzzy množina* (alebo *fuzzy podmnožina*) F na množine objektov Ω je definovaná ako výsledok zobrazenia:

$$\mu_F : \Omega \rightarrow [0,1]$$

pre $\forall x \in \Omega$, $\mu(x)$ *je stupeň pravdivosti* (tiež *stupeň členstva*) *x do F.*

Definícia (negácia): Funkcia $n : [0,1] \rightarrow [0,1]$ ak je nerastúca a platí $n(0) = 1$ a $n(1) = 0$. Negácia n je *ostrá negácia*, ak je ostro klesajúca a spojitá.

Definícia (t-norm): Funkcia $T : [0,1]^2 \rightarrow [0,1]$ je *t-norm* (*triangulárna norma*), ak platia nasledujúce štyri podmienky:

- Ekvivalentná podmienka $T(1, x) = x \quad \forall x \in [0,1]$
- T je komutatívne $T(x, y) = T(y, x) \quad \forall x, y \in [0,1]$
- T je neklesajúce v oboch prvkoch $T(x, y) \leq T(u, v)$ pre všetky $0 \leq x \leq u \leq 1$ a $0 \leq y \leq v \leq 1$
- T je asociatívna $T(x, T(y, z)) = T(T(x, y), z) \quad \forall x, y, z \in [0,1]$

Definícia (t-conorm): Funkcia $S : [0,1]^2 \rightarrow [0,1]$ je *t-conorm* (tiež *s-norm*), ak platia nasledujúce štyri podmienky:

- Ekvivalentná podmienka $S(0, x) = x \quad \forall x \in [0,1]$
- S je komutatívne $S(x, y) = S(y, x) \quad \forall x, y \in [0,1]$
- S je neklesajúce v oboch argumentoch $S(x, y) \leq S(u, v)$ pre všetky $0 \leq x \leq u \leq 1$ a

$0 \le y \le v \le 1$

- S je asociatívna $S(x, S(y, z)) = S(S(x, y), z) \quad \forall x, y, z \in [0,1]$

T-norm(y) a t-conorm(y) sú spájané do dvojíc a pre vhodný operátor negácie je možné zovšeobecnenie pomocou De Morganovho zákona.

Definícia (De Morganova trojica): Nech T je t-norm a S je t-conorm a n je ostrá negácia. Potom $\langle T, S, n \rangle$ je *De Morganova trojica* práve vtedy ak platí

$$n(S(x, y)) = T(n(x), n(y))$$

Funkcia T (t-norm) a z nej (pomocou De Morganovej trojice) odvodená funkcia S (t-conorm) tvoria základné operácie v teórii fuzzy logiky. Dvojica funkcií, ktorá splňuje vlastnosti t-norm a t-conorm bude označená @ (tiež *agregačný operátor*). V závislosti na použitej funkcii t-norm sa často hovorí ako *t-norm fuzzy logike.*

Názov	**t-norm**	**t-conorm**
Zadeh	$min(x, y)$	$max(x, y)$
pravdepodobnostná	$x*y$	$x + y - xy$
Lukasiewicz	$max(x + y - 1, 0)$	$min(x + y, 1)$
Hamacher ($\gamma > 0$)	$(xy) / (\gamma + (1 - \gamma)(x + y - xy))$	$(x + y + xy - (1 - \gamma)xy) / (1 - (1 - \gamma)xy)$
Yager ($p > 0$)	$max(1 - ((1 - x)^p + (1 - y)^p)^{1/p}, 0)$	$min((x^p + y^p)^{1/p}, 1)$
Weber ($\lambda > -1$)	$max((x + y - 1 + \lambda xy)/(1 + \lambda), 0)$	$min(x + y + \lambda xy, 1)$
drastický	x ak $y = 1$ y ak $x = 1$ inak 0	x ak $y = 0$ y ak $x = 0$ inak 1

Dá sa povedať, t-norm reprezentuje logický & a t-conorm logické $\vee$. Pre úplnosť je zadefinovaná ekvivalencia, ako jedna z najpoužívanejších relácii.

Definícia (Ekvivalentná relácia): Funkcia $E : [0,1]^2 \to [0,1]$ je *ekvivalencia* ak spĺňa nasledujúce podmienky:

i. $E(x, y) = E(y, x)$ pre $\forall x, y \in [0,1]$

ii. $E(0,1) = E(1,0) = 0$

iii. $E(x, x) = 1$ pre $\forall x \in [0,1]$

iv. $x \le x' \le y' \le y \Rightarrow E(x, y) \le E(x', y')$

2.4 Pravdepodobnosť a štatistika

Nech (Ω, $\boldsymbol{A}$) je merateľný priestor. Prvky množiny Ω budú nazývané *elementárne javy* a značené ako ω, prvky σ-algebry $\boldsymbol{A}$ budú nazývané *javy* a značené veľkým písmenom zo začiatku abecedy.

Definícia (Pravdepodobnosť): Pravdepodobnosť P je definovaná ako miera na $\boldsymbol{A}$ s vlastnosťou $P(\Omega) = 1$, to znamená P je množinová funkcia na $\boldsymbol{A}$ s vlastnosťami: A

i. $P(A) \geq 0, \quad A \in \boldsymbol{A}$

ii. $P(\Omega) = 1, \quad P(\emptyset) = 0$

iii. $P(\cup_{n=1}^{\infty} A_n) = \sum_{n=1}^{\infty} P(A_n)$, ak je $\{A_n\}$ postupnosť po dvoch disjunktných javoch.

Trojica (Ω, $\boldsymbol{A}$, P) je nazývaná *pravdepodobnostný priestor*. Ďalej množina Ω je množina všetkých možných výsledkov (*istých javov*), $\emptyset$ (prázdna množina) je *nemožný jav* a ak $\omega \in A$ potom *nastal jav A*.

Definícia (Podmienená pravdepodobnosť): Nech $A, B \in A, \mathrm{P}(B) > 0$. *Podmienená pravdepodobnosť javu A za podmienky B* je definovaná vzťahom

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

Veta (O násobení pravdepodobností): Pre ľubovolných $n + 1$ javov $A_0, A_1, \ldots, A_n$ takých, že $P(A_0 A_1 \ldots A_{n-1}) > 0$, platí

$$P(A_0 A_1 \ldots A_n) = P(A_0) P(A_1|A_0) \ldots P(A_n|A_0 A_1 \ldots A_{n-1})$$

Veta (O celkovej pravdepodobnosti): Ak je $P(\cup_n B_n) = 1$ kde $\{B_n\}$ je konečna alebo spočetná postupnosť vzájomne sa vylučujúcich javov, ak je $(B_n) > 0$ pre všetky n a ak je $A \in \boldsymbol{A}$ potom $P(A) = \sum_n P(A|B_n) P(B_n)$

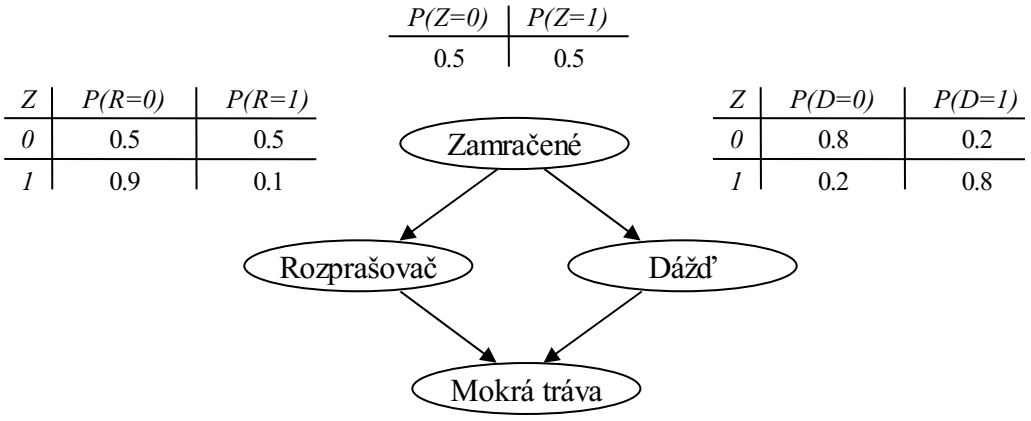
Veta (Bayesova): (tiež Bayesov teorém alebo zákon) Za predpokladu predchádzajúcej vety a predpokladu $P(A) > 0$ platí

$$P(B_m|A) = \frac{P(A|B_m) P(B_m)}{\sum_n P(A|B_n) P(B_n)} \quad \text{pre všetky } m$$

Javy A, B sa nazývajú *nezávislé javy*, ak $P(A \cap B) = P(A) P(B)$ v opačnom prípade sú to *závislé javy*.

2.5 Bayesova sieť

Bayesove siete budú ukázané na obecne používanom príklade mokrého trávnika. V tomto príklade môže byť trávnik mokrý z dôvodu dažďa alebo z dôvodu zapnutého rozprašovača. Inými slovami to či je trávnik mokrý, závisí od toho, či pršalo alebo či bol zapnutý rozprašovač. Ďalej sa predpokladá, že dni môžu byť zamračené alebo jasné. Pravdepodobnosť, že bude pršať v zamračený deň je vyššia ako v deň jasný. Podobne je pravdepodobnejšie, že rozprašovač je zapnutý počas jasného dňa ako počas zamračeného. Obe pravdepodobnosti (dážď a zapnutý rozprašovač) sú podmienené zamračeným počasím. Celá situácia bude modelovaná jednoduchou Bayesovou sieťou obsahujúcou štyri binárne pravdepodobnostné premennými *Zamračené* (*Z*), *Rozprašovač* (*R*), *Dážď* (*D*) *a Mokrá tráva* (*M*), ktoré môžu byť v stave 1 (pravda, *true*) alebo 0 (nepravda, *false*). Závislosť pravdepodobností od predchádzajúcich udalostí znázorňujú orientované hrany.



P(Z=0)	*P(Z=1)*
0.5	0.5

Z	*P(R=0)*	*P(R=1)*
0	0.5	0.5
1	0.9	0.1

Z	*P(D=0)*	*P(D=1)*
0	0.8	0.2
1	0.2	0.8

R	*D*	*P(M=0)*	*P(M=1)*
0	*0*	1.0	0.0
1	*0*	0.1	0.9
0	*1*	0.1	0.9
1	*1*	0.01	0.99

Obrázok 2.1, Jednoduchá Bayesova sieť znázorňujúca podmienené pravdepodobnosti mokrej trávy od nezávislých udalostí zapnutého rozprašovača a dažďa

Pre vrchol grafu je špecifikované podmienené pravdepodobnostné rozdelenie. V tomto príklade sa o všetkých premenných uvažuje ako o diskrétnych premenných, preto je možné podmienenú pravdepodobnosť jednotlivých vrcholov grafu zapísať pomocou podmienenej pravdepodobnostnej tabuľky. Tabuľky obsahujú pravdepodobnosti, pre jednu zo všetkých možných hodnôt čiže, pre kombináciu hodnôt ich rodičov, teda pravdepodobnosť, že tráva je mokrá, za predpokladu, že nepršalo ale bol zapnutý rozprašovač, je $P(M=1 \mid R=1,D=0)=0,9$.

Pre potreby Bayesových sieti bude zavedená notácia. Ak existuje orientované spojenie z uzlu *A* do uzlu *B* potom uzol *A* bude nazývaný *rodičom uzlu B* a uzol *B* bude nazývané *potomkom uzlu A*. Ak uzol *A* nemá žiadnych rodičov bude nazývaný *nepodmieneným uzlom* inak *podmieneným uzlom*[1]. Z vety o násobení pravdepodobností, je pre uvedený príklad možné odvodiť združenú pravdepodobnosť P_N ako

$$P_N(Z, R, D, M) = P(Z) * P(R \mid Z) * P(D \mid Z, R) * P(M \mid Z, R, D)$$

Po úprave, keďže *D* je nezávislé od *R* pri jeho rodičovi *Z* a *M* je nezávislé od *Z* pri jeho rodičoch *R* a *D*, je možné vzťah pre výpočet združenej pravdepodobnosti bayesovej siete zjednodušiť na $P_N(Z, R, D, M) = P(Z) * P(R \mid Z) * P(D \mid Z) * P(M \mid R, D)$ Z toho vyplýva, že v prípade *n* binárnych vrcholov je miesto potrebné na reprezentovanie $O(n * 2^k)$, kde *k* je maximálny počet rodičov vrcholu, inak je potrebné $O(2^n)$ miesta na reprezentovanie. Toto umožňuje jednoduchšie učenie siete a rýchlejší beh algoritmov, ktoré môžu skončiť v lineárnom čase (na základe počtu hrán) namiesto exponenciálneho času (na základe počtu parametrov).

Definícia (Bayesova sieť): Bayesovou sieťou nad množinou *U* bude nazývaná dvojica (*G*, *P*), kde *G* sú vrcholy orientovaného acyklického grafu a *P* je *množina pravdepodobnostných tabuliek*.

Množinou pravdepodobnostných tabuliek sa rozumie pre každé $A \in U$ samostatná podmienená pravdepodobnostná tabuľka (*conditional probability table – cpt*) $P(A \mid parent(A))$, kde $parent(A)$ predstavuje rodičov *A* v grafe *G*. Ďalej nad množinou *U* môžme dopočítať

$$P(A_1, \ldots, A_n) = \prod_{i=1}^{n} P(A_i \mid parent(A_i))$$

Poznámka v Bayesovej sieti je *Markov blanket* nejakého vrcholu *A* množina vrcholov zložená z rodiča vrcholu *A*, potomkov vrcholu *A* a ich rodičov. Pomocou Bayesovej vety je možné odvodiť, že pravdepodobnosť, že trávnik je mokrý z dôvodu, že bol zapnutý rozprašovač je

$$P(R=1 \mid M=1) = \frac{P(R=1) * P(M=1 \mid R=1)}{P(M=1)} = \frac{P(R=1, M=1)}{P(M=1)} = \frac{0{,}2781}{0{,}6471} = 0{,}4298$$

podobne by to vyzeralo aj pre dážď

1 Uncondition vs. Conditional nodes

2.6 Markovove Siete

Markovova sieť (tiež Markovove náhodne polia) je model pre združené rozdelenie množiny premenných $X = (X_1, X_2, \ldots, X_n) \in \boldsymbol{X}$. Je zložená neorientovaného grafu *G* a množiny potencionálnych funkcií φ_k . Graf má vrcholy pre každú premennú a celkový model grafu má potencionálnu funkciu pre každú kliku v grafe. Potencionálna funkcia je nezáporná reálna funkcia stavov odpovedajúcej kliky. Združené rozdelenie reprezentované Markovovou sieťou je dané

$$P(X=x) = \frac{1}{Z}\prod_k \varphi_k(x_{\{k\}})$$

kde x{k} stav k-tej kliky (stav premenných obsiahnutých v klike). *Z* , tiež *rozdeľovacia funkcia* (*partition funcion*) je daná ako $Z = \sum_{x\in \boldsymbol{X}} \prod_k \varphi_k(x_{\{k\}})$. Markovove siete sú často bežne reprezentované ako log-lineárne modely, s každým potenciálom kliky nahradeným exponentom vážených súm vlastnosti stavov

$$P(X=x) = \frac{1}{Z}\exp(\sum_j w_j f_j(x))$$

Vlastnosťou môže byť každá reálna funkcia stavov. V tomto dokumente, a pre jednoduchosť, bude táto vlastnosť binárna $f_i \in \{0,1\}$. Vo väčšine priamych prekladov z potenciálnej funkcie je jedna vlastnosť odpovedajúca pre každý možný stav $x_{\{k\}}$ každej kliky, s váhou $\log \varphi_k(x_{\{k\}})$. Reprezentácia je exponenciálna v závislosti na veľkosti kliky. Poznámka v Markovovej sieti je *Markov blanket* nejakého vrcholu *A* množina všetkých jeho susedov.

3 Preferencie obecne

Užívateľskou preferenciou sa rozumie čokoľvek čo užívateľ preferuje respektíve uprednostňuje. *Modelom užívateľských preferencií* je spôsob akým sú jednotlivé užívateľské preferencie chápané, uložené respektíve interpretované. Táto diplomová práca patrí medzi porovnávacie diplomové práce. Hlavnou témou je zostaviť prehľad súčasných modelov užívateľských preferencií a ich porovnanie. Ďalšou úloh je rozobratie problematiky užívateľských preferencií z pohľadu psychológie jednotlivých užívateľov. Súčasťou je aj navrhnutie kritérií hodnotenia jednotlivých modelov v rôznych oblastiach využitia.

3.1 Sémantický web

Jednou z oblasti využívania modelov užívateľských preferencií je v problematike sémantického webu. Ideu *sémantického webu* prvý krát prezentoval v máji 2001 Tim Berners-Lee [33], tvorca súčastného webu a riaditeľ konzorcia W3C, v Scientific American. Upozornil na skutočnosť, že súčasná sieť WWW je v podstate len rastúce množstvo webových stránok, v ktorom je stále zložitejšie nájsť relevantné informácie. Samotná myšlienka sémantického webu spočíva, v pridaní významu, k informáciám, v strojovo čitateľnej podobe. Predstava spočíva v tom, že počítač obsahuje isté informácie o užívateľovi, ako napríklad zaujmi, informácie o zamestnaní (plat, činnosť, . . .), bydlisko, rodinné záväzky, denný (týždenný) program, . . . Ak sa potom užívateľ rozhodne kúpiť napríklad auto ako prvé mu budú ponúknuté autá v jeho finančnej kategórii, prípadne len značky, ktorých predajne sú v blízkosti jeho bydliska. Iným príkladom môže byť hľadanie vhodného autoservisu. V tomto prípade sú ponúknuté najbližšie autoservisy, ale samozrejme také, ktoré budú mať otvorené a nebudú obsadené v užívateľovom voľnom čase. Existuje tu istá asociácia, jednak medzi jednotlivými informáciami prístupnými na internete a jednak medzi informáciami o užívateľoch. Ďalším príkladom môže byť vyhľadávanie na internete, kde z množiny všetkých prijateľných možností sú najprv ponúknuté tie, ktoré majú najbližšie k záujmom užívateľa, zamestnaniu prípadne obore pôsobenia. Hlavné problémy pri aplikácii sémantického webu v praxi je možné rozdeliť na:

- zhromaždenie informácii dostupných na internete (datamining) v závislosti na jazykoch (ontológia)

- zaradenie získaných informácií a interpretácia významu dát (RDF[1])
- rozhodnutie, o ktorú oblasť má užívateľ záujem (preferencie)

Problém rozhodnutia o preferenciách užívateľa je pomerne komplikovaný, keďže v súčasnom svete málo ktorý užívateľ poskytne o sebe, na jednej strane, príliš osobné údaje a na druhej strane údaje, ktoré poskytne nemusia byť presné, kompletné ani postačujúce. Tento problém je možné riešiť postupným učením z už urobených rozhodnutí a analýzou nových rozhodnutí. Následne pomocou takto získaných informácií potom vytvoriť a rozširovať model predstavujúci užívateľské preferencie. Existuje viacero možností ako modelovať užívateľské preferencie a práve touto problematikou sa zaoberá táto diplomová práca.

Iný spôsob k využitiu modelov užívateľských preferencií, bez dostupnosti sémantického webu, je založený na zbieraní rozhodnutí (preferencií) jednotlivých užívateľov a ich následná aplikácia v iných oblastiach. Vhodným príkladom je internetový obchod s registrovanými užívateľmi a širokou škálou poskytovaných produktov. Ak sa nejaký užívateľ rozhodol pre notebook s väčšou odolnosťou voči nárazom a s batériou, ktorá dlhšie vydrží, ale zároveň pre mobil s veľkým rozlíšením. Dá sa predpokladať, že aj iný užívateľ, ktorý si už podobný cestovný notebook zakúpil a teraz má záujem o telefón bude preferovať displej s veľkým rozlíšením, preto by mu mal byť ponúknutý ako prvý. V podstate sú všetci zákazníci rozdelený do kategórií respektíve skupín podľa ich preferencií. Pri dlhodobom zaznamenávaní týchto preferencií je možné sledovať aj najmenšie zmeny preferencií jednotlivých skupín a podľa toho potom informovať ostatných členov v tejto preferenčnej skupine, o výrobkoch, o ktoré budú mať pravdepodobne tiež záujem. Iná výhoda je založená na myšlienke, že ak niekto poskytne informácie o svojej súčasnej situácii, ako napríklad študent informatik, muž, zaujmi šport, hudba, počítač . . . aj ostatný členovia rovnakej preferenčnej skupiny budú mať najskôr niečo spoločne s týmto študentom. V prípade problému firmy, ktorá vyrába textil a chce prieskumom zistiť, na aký druh textilu sa má zamerať by samozrejme najradšej oslovila skupinu ľudí, ktorý u nej nakupujú najviac. Riešením je nájdenie najväčších skupín s rovnakými preferenciami. Teda zistenie o akých ľudí sa v reálnom živote jedná a nakoniec oslovenie tejto skupiny ľudí a zistenie ich názoru.

Vo všeobecnosti je možné povedať, že hlavnou myšlienkou a cieľom sémantického webu je usporiadanie objektov od najlepšieho po najhorší, v závislosti na preferenciách

1 Resource Description Framework

užívateľa. Tento text sa zaoberá rozdielnymi reprezentáciami (modelmi) užívateľských preferencií a prácou s nimi.

3.2 Vplyvy na preferencie

Na preferencie konkrétneho užívateľa vplýva veľké množstvo faktorov. Tieto faktory je možné rozdeliť podľa rôznych uhlov pohľadu. Najprv je to samotná motivácia respektíve potreba (chcem nový počítač). Ďalšou fázou je istá vnútorná neistota pri rozhodovaní (potrebujem nový počítač?) a nakoniec zmeny v názoroch podľa novo zistených informácií (najprv preferujem dobrú grafiku, potom radšej dobrý procesor). Z tohoto pohľadu je možné vplyvy na jednotlivé fázy vo vývoji preferencií rozdeliť na:

- vonkajšie podmienky (počasie, reklama, ponuka trhu, . . .)
- vnútorné podmienky (osobné potreby, záujmy, zameranie, . . .)
- obmedzujúce podmienky (prostredie, finančná situácia, zdravotný a rodinný stav, . . .)

Na vonkajšie podmienky je možné dívať sa ako na niečo globálne a neovplyvniteľné, ale na druhej strane ovplyvňujúce samotných užívateľov. Obmedzujúce podmienky budú chápané ako vplyvy na samotné preferencie, ale meniace sa v závislosti na čase (zmena prostredia, finančnej situácie, rodinného stavu, . . ., vyvolá zmenu preferencií). Čo sa týka vnútorných podmienok je ich ďalším rozdelením to, či užívateľ uprednostňuje konkrétny objekt (auto pred motorkou, stolový počítač pred notebookom), alebo konkrétne vlastnosti (atribúty) objektu (procesor pred grafikou, bielu farbu pred červenou farbou). V tomto rozdelený preferencií užívateľovi:

- záleží na vlastnostiach objektu
- nezáleží na vlastnostiach objektu

Ako už bolo spomenuté v úvode m*odelom užívateľských preferencií* sa nazýva spôsob ohodnotenia objektov užívateľom respektíve ich samotná interpretácia. Preto je možné jednotlivé modely hodnotiť podľa toho či:

- berú, alebo neberú v úvahu atribúty objektov a ich rozdielnu dôležitosť
- umožňujú interpretáciu vonkajších podmienok a aplikovať ich pri rozhodovaní
- dokážu sa meniť v závislosti na čase (podľa zmien preferencií užívateľa)

Otázkou ostáva, ako vytvoriť model užívateľských preferencií. Vo všeobecnosti existuje množstvo prístupov, ale takmer všetky vyžadujú históriu rozhodnutí predchádzajúcich užívateľov. Jednou z možností je, nechať užívateľa vyplniť dotazník o jeho záujmoch, zameraní, osobné údaje a samozrejme finančnú situáciu teba príjem. Podľa týchto údajov potom nájsť podobného užívateľa a ním zvolené a preferované objekty a tie potom ponúknuť aj novému užívateľovi. Preferencie nového užívateľa potom zaznamenávať do histórie k presnejším budúcim výsledkom.

Druha možnosť je formou ohodnocovania jednotlivých objektov a podľa týchto ohodnotení, viacerými užívateľmi, rozdeliť celú množinu produktov, do logicky súvisiacich, skupín. Novému užívateľovi budú najprv ponúknuté produkty reprezentujúce každú skupinu. Podľa toho pre aký produkt sa rozhodol mu budú ponúkané aj ostatné produkty z tejto skupiny.

Treťou z možností je určenie preferencií podľa špecifikácie produktu. História preferencií si pamätať jednotlivé špecifikácie a aj to, pre aký produkt sa užívateľ nakoniec rozhodol. Napríklad laik užívateľ sa rozhodne pre kúpu notebooku, jeho špecifikácia produktu bude veľmi nepresná, ale ak sa nakoniec pre nejaký notebook rozhodne bude pravdepodobne, že aj iný užívateľ s podobou (nepresnou, neodbornou) špecifikáciou bude mať rovnaké požiadavky. Pre zhrnutie modelovanie preferencií môže byť založené na základe:

- informáciach o užívateľoch
- ohodnocovaní objektov
- špecifikovaní preferencií

Tento text je zameraný na kombináciu medzi informáciami o užívateľoch spolu s prípadnou špecifikáciou konkrétnych (ale nie všetkých) preferencií a nakoniec zoradenie množiny najvhodnejších výsledkov na základe ohodnotenia.

4 Modely užívateľských preferencií

Ako už bolo povedané v predchádzajúcej kapitole, modelom užívateľských preferencií sa rozumie konkrétny spôsob reprezentácie reprezentácie preferencií. V tejto kapitole bude popísané rozdelenie modelov na základe tohto rozdelenia budú podrobne predstavené niektoré modely.

4.1 Rozdelenie modelov

Keďže existuje viacero pohľadov podľa ktorých by mohli byť rozdelené, samotné delenie modelov do skupín je pomerne zložité. Prvý spôsob rozdelenie je na graficky a dôkazový, čo predstavuje či model vychádza z grafickej štruktúry alebo z dôkazu, že niečo platí. Iným pohľadom na rozdelenie je či model vychádza z teórie pravdepodobnosti alebo teórie fuzzy logiky.

4.1.1 Graficky a dôkazový prístup

Vo všeobecnosti existujú dva prístupy (pohľady) na logické programy *modelový* (*graficky*) a *dôkazový.* Príklad mokrého trávnika predstavuje typický príklad modelového pohľadu, ktorý obsahuje štyri možnosti {Zamračené, Rozprašovač, Dážď, Mokrý trávnik} tieto možnosti sú tiež nazývané *Herbrandovo univerzum* (respektíve *báza*). Je zložená zo všetkých skutočností, ktoré je možné konštruovať z predikátových, konštantných a funkčných symbolov programu. Herbrandovo univerzum (v istom zmysle) určuje množinu všetkých možných situácii vo svete popisovaných programom. V príklade s trávnikom, existuje $2^4 = 16$ možných priradení pravdivostných hodnôt (v klasickej dvoj-hodnotovej logike) do Herbradovho univerza. Tieto priradenie sú tiež nazývané Herbrandova interpretácia a zachycujú popísanie situácií vo svete. Logický program založený na modelovom prístupe obmedzuje množinu všetkých situácií len na tie, ktoré sú možné. Teda najdôležitejšou vecou v modelovom prístupe je to, že špecifikuje, ktorá časť vo svete je možná vzhľadom k použitej logickej teórii.

Iný spôsob pohľadu na logické programy pochádza z dôkazovej teórie. Z tejto perspektívy môže byť logický program použitý k poskytnutiu informácie o tom, že niektoré fakty, pravidlá a dotazy sú logicky spojené s programom. Ako príklad môže byť zobraná funkcia kompilátora k určeniu syntaktickej správnosti jednotlivých častí programu prípadne podobná funkcia bez-kontextových gramatík. V týchto situáciach je dôležitá

pravdivosť zadaného vstupu, takže odpoveďou je buď true alebo false. Dôkaz je typicky konštruovaný pomocou SLD-rezolúcie. Viac formálne, pre daný dotaz $:-G_{1,}...,G_n$ a pravidlo $G:-L_{1,}...,L_n$ také, kde $G_1\theta=G\theta$, bude použití rezolučný dôkaz v dotaze $:-L_1\theta,...,L_n\theta,G_2\theta,...,G_n\theta$. Úspešné rezolučné vyvrátenie dotazu $:-G$ je potom postupnosť rezolučných krokov, ktoré vedú k prázdnemu dotazu $:-.$, zlyhanie nekonči prázdnym dotazom. Tiež je dôležite uvedomiť súvislosť medzi SLD-deriváciu a tradičnou deriváciou používanou v bez-kontextových gramatikách. V oboch prípadoch však existuje mapovanie jedna k jednej medzi týmito dvoma pohľadmi na gramatiky.

Existujú teda dva spôsoby, ako sa pozerať na logiky program. Prvý pohľad, je modelový. Závisí na tom, čo z reálneho sveta je možné interpretovať tak aby to uspokojilo potreby programu sem patria modely založené na teórii Bayesových a Markovových sietí. Druhý pohľad je dôkazový. Ten určuje, čo je možné dokázať vzhľadom na daný cieľ. Samozrejme, obe z týchto pohľadov sú úzko spojene, pretože logický program má rezolučné vyvrátenie, len v prípade, že patrí do Herbrandoveho modelu programu.

4.1.2 Fuzzy logika a Pravdepodobnosť

Rozdiel medzi teóriou pravdepodobnosti a fuzzy logikov je pomerne jasný Z pravdepodobnosti plynie aká je pravdepodobnosť, že sa niečo stane. Fuzzy logika na druhej strane poskytuje širší priestor na ohodnotenia nejakých objektov vo všeobecnosti tiež preferencií. Z iného pohľadu odpoveď na otázku „aká je pravdepodobnosť, že je niečo dobré na základe určitých predpokladov“ z pohľadu fuzzy by mohlo znieť ako (ak dobré má hodnotu 1) „niečo je dobre na 0,8 tiež na základe určitých predpokladov“. To znamená ak existuje niečo lepšie tak buď je to pravdepodobnejšie alebo to má lepšiu hodnotu.

4.2 Prehľad modelov

Jednotlivé modely je možné rozdeliť podľa teórií, z ktorých vznikli a z ktorých vychádzajú. Medzi modely neobsahujúce logické programovanie je možné zaradiť dvoj-hodnotový model, ako jeden z najjednoduchších modelov. K ostatným druhom modelov je nutné povedať, že vznikli kombináciou (minimálne) dvoch teórií, z ktorých jednu poväčšine predstavuje logické programovanie respektíve predikátová logika prvého rádu.

K modelom založeným na grafickom pohľade patria modely vychádzajúce z kombinácie Bayesových sietí a logického programovania, obecne nazývané Knowledge-Based Model Construction (ďalej KBMC) [36]. K najrozšírenejším predstaviteľom tejto

skupiny patrí Bayesovo Logické Programovanie (BLP) [14][13][12], ale tiež Probabilistic Logic Programs (PLP) [22][21].

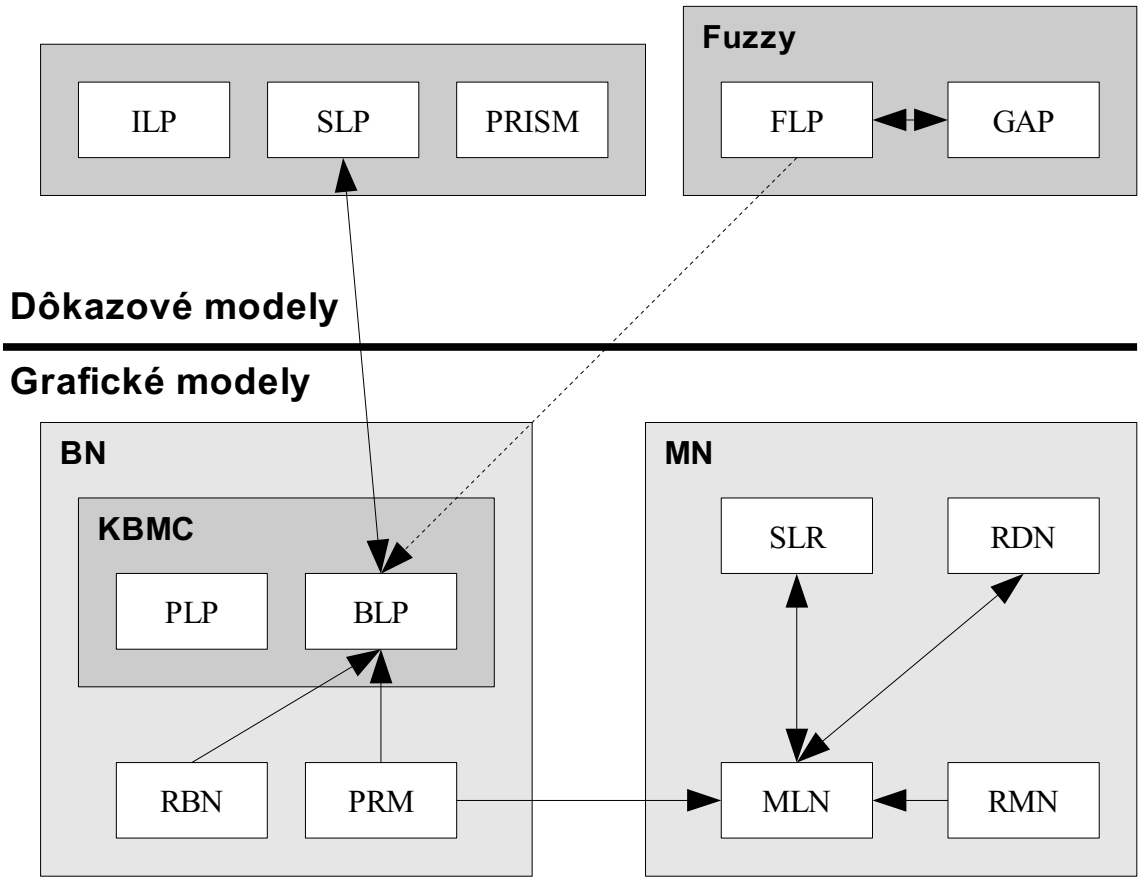
Z trocha iného uhla pohľadu vychádza Probabilistic Relational Model (PRM) [6][8] [7][5], ktorý je kombináciou frame-based systémov a Bayesových sieti. Tento model bol rozšírený o Entity-Relationship model (ER-model) tým vznikol istý jazyk pre pravdepodobnostný ER model [10]. Je nutné poznamenať, že ER-model je vlastne špeciálnym prípadom logiky, teda pravdepodobnostný ER-model pripúšťa logické výrazy ako obmedzovania nato, ako základnú sieť konštruovať, ale pravdivostná hodnota týchto výrazov musí byť známa vopred.

Ďalším príkladom sú modely vychádzajúce z kombinácie Makovových sietí a predikátovej logiky prvého rádu, kde patri Markové Logické Siete (MLN) [3][2]. Obdobne z Markovových sietí vychádza aj model Relational Markov Networks (RMN) [31][32], ktorý používa databázové dotazy ako šablóny pre kliku a majú funkciu pre každý stav kliky. MLN zovšeobecňujú RMN tým, že poskytujú silnejší jazyk pre výstavbu funkcií (logika prvého rádu namiesto konjunktívnych dotazov), a umožňuje neistotu cez vzťahy medzi atribútmi (nie len atribúty jednotlivých objektov). RMN sú exponenciálne vo veľkosti kliky, zatiaľ čo MLN umožňuje užívateľovi zistiť počet prvkov, aby bola možná mierka do omnoho väčšej veľkosti kliky. Ďalej sem patrí Relational Dependency Network (RDN) [20], v ktorom každý uzol pravdepodobnostne podmienený jeho Markov blanket je daný rozhodovacím stromom. Každá RDN korešponduje s MLN a naopak, danú pevnou distribúciou Gibbs sampler operating [9].

K modelom založených na dôkazovej teórii patria modely vychádzajúce z kombinácie logického programovania a log-lineárneho modelu Stochastické Logické Programy (SLP) [19][1], v podstate sa jedna rozšírenie stochastických bez-kontextových gramatík. V zásade podobne modely sú Independent Choice Logic (ICL) [24][23] a PRogramming In Statistical Modeling (PRISM) [28][29]. Do tejto skupiny je ďalej možné zaradiť Probabilistic Constraint Logic Programming (P-CLP) [27].

Samostatnú skupinu predstavuje Fuzzy Logické Programovanie (FLP) [34][26] vychádzajúce z rozšírenia logického programovania o fuzzy (viac-hodnotovej) logiky. Taktiež, podobne ako SLP a PRISM patrí do skupiny modelov založených na dôkazovej teórii. Medzi modely vychádzajúce z fuzzy logiky patria aj zovšeobecnené anotované programy (Generalized Annotated Programs - GAP) [15][11], tiež zovšeobecnené anotované programy. GAP pozostáva z Hornových klauzúl logického programovania

zložených len z anotovaných atómov (atómom, ktoré máju naviac priradenú hodnotu z intervale [0, 1]).



Obrázok 4.1, Prehľad modelov a závislostí medzi nimi

Medzi jednotlivými modelmi už existujú dokázané ekvivalencie či už na základe vhodnej substitúcie alebo len zmeny interpretácie. Ekvivalencia medzi SLP a BLP bola ukázaná v [25]preto sa model SLP v tomto nerozoberá ale je nahradený modelom BLP.

Iný pohľad na preferencie je ukázaný v [30], ktorý vychádza zo štatisticky. Na základe volených objektov užívateľom vytvára vektor vlastností, ktoré majú zvolené objekty podobné. Následne ponúka ďalšie objekty obsahujúce len vektor týchto vlastností. Ako príklad môže byť uvedené vyhľadávanie v dokumentoch, kde vlastnosti sú reprezentované slovami (alebo kombináciou slov) a objekty sú jednotlivé dokumenty.

V tejto diplomovej práci budú podrobne rozobrané modely FLP pre viac-hodnotovú logiku v logickom programovaní založený na dôkaze, BLP využívajúci Bayesove siete, na druhej strane MLN využívajúci Markovove siete a nakoniec PRM s jeho rozšírením ER-modelu.

4.3 Dvoj-hodnotový model

Najjednoduchším modelom užívateľských preferencií je dvoj-hodnotový model založený na ohodnocovaní objektov pomocou klasickej binárnej logiky. V tejto časti bude použitá notácia $u_1, \ldots, u_k \in U$ pre množinu užívateľov U ďalej $o_1, \ldots, o_n \in O$ pre množinu objektov O a $a^i_1, \ldots, a^i_j \in A^i$ pre množinu atribútov A^i objektu o^i. V tomto modele sa jednotlive objekty ohodnocujú jednou z dvoch možností $\{0, 1\}$ (áno/nie prípadne podľa potreby pravda/nepravda alebo páči/nepáči). Výsledkom je, pre každého užívateľa, vektor veľkosti n (počet objektov). Obecne sa jedná o zobrazenie $v(u)$: $O \rightarrow [0, 1]^n$, kde v je hodnotenie užívateľom u, ktoré množine objektov O priradí vektor $[0, 1]^n$.

K určeniu preferencií jednotlivých užívateľov sa požíva metóda *kolaboratívneho filtrovania*, ktorej hlavná myšlienka je založená na podobnosti preferencií medzi jednotlivými užívateľmi. K určeniu podobnosti preferencií $s(u_1, u_2)$ dvoch rozdielnych užívateľov u_1 a u_2 je možné použiť jeden zo vzorcov:

$$s(u_1, u_2) = \sum_{i=1}^{n} \frac{(v(u_1) \text{ and } v(u_2))[i]}{n} \quad \text{alebo} \quad s(u_1, u_2) = 1 - \sum_{i=1}^{n} \frac{(v(u_1) \text{ xor } v(u_2))[i]}{n}$$

Zo vzorcov vyplýva, že najväčšia podobnosť je pre najväčšiu hodnotu s. Takže užívateľovi u_1 budú ponúknuté objekty preferované užívateľom u_2 a naopak.

Modelovým príkladom pre dvoj-hodnotový model môže byť ohodnocovanie obrazov, kde sú užívateľovi ponúkané obrazy, ktoré ohodnocuje hodnotami 1 pre páči a 0 pre nepáči. Pomocou metódy kolaboratívneho filtrovania sa s každým ďalším ohodnoteným obrazom zvyšuje pravdepodobnosť, že sa ponúknutý obraz bude užívateľovi páčiť.

Tento model je možné rozšíriť o tretiu hodnotu (napríklad 2) pre prípady, v ktorých užívateľ nevie alebo nechce hodnotiť.

4.4 Fuzzy logické programovanie

Fuzzy logické programovanie je spojenie fuzzy (viac-hodnotovej) logiky a logického programovania. Pre motiváciu samotné fuzzy logické programovanie pozostáva z Hornových klauzúl (faktov a pravidiel) logického programovania, ku ktorým je pridaná zobrazenie do intervalu [0,1] (*confidence factor - cf*) z fuzzy logiky, teda fuzzy funkcia. K nájdeniu odpovede na nejaký dotaz je štandardne použitá SLD-rezolúcia z logického

programovania, ale k dosiahnutiu výslednej hodnoty (pravdivosti dotazu) je použitá teória fuzzy logiky a jej rozšírenie o ďalšie logické spojky, ktoré predstavujú ľubovolné funkcie, ktoré splňujú vlastnosti t-normy. V prípade induktívneho programovania sa k prehľadávaní priestoru klauzúl používajú operátory LG (*least generalisation*) a GS (*greatest specialisation*). LG operátor sa používa na výpočet najmenšej generalizácie, na druhej strane GS operátor sa používa na najväčšiu špecializáciu klauzúl vstupnej množiny. Pravidlo fuzzy logického programu je formula tvaru $\varphi \leftarrow_i \psi$, kde ψ je formula jazyka L (jazyka predikátovej logiky) bez kvantifikátorov φ je atomická formula. Formula ψ sa nazýva *telo* pravidla. Formula φ sa nazýva hlava pravidla. Fakt fuzzy logického programu je ľubovoľná atomická formula. Fuzzy logický program P je zobrazenie definované na konečnej množine faktov a pravidiel do množiny pravdivostných hodnôt $V \in [0,1]$. Teda fuzzy logický program je teória v jazyku L s dodatočnou množinou spojok $\{\leftarrow_i :$ kde i patrí do množiny t-norm $\}$ [1]. Jedná sa teda o Hornové klauzule, ale namiesto klasických spojok predikátovej logiky obsahujú viachodnotové spojky fuzzy logiky

Definícia (Správna odpoveď): Nech P je fuzzy logický program a $q \in PF$ je dotaz, kde PF je množina prvotných formúl. Potom $x \in [0,1]$ sa nazýva *správna odpoveď* , ak pre každé $I : PF \rightarrow [0,1]$ platí:

$$\text{ak } I \models \text{P, potom } I(q) \geq x \quad .$$

Z tejto definície vyplýva, že nulová odpoveď je vždy správna ale v podstate nič nehovorí. Teda je potrebné nájsť $\sup\{I(q) : I \models P\}$.

Definícia (Výpočtový krok): Nech $? - q_0, q_1, \ldots, q_n$ je množina dotazov, kde $q_i \in PF \; pre \; každe \; i$ a q_j je vybraný atóm a $q_j \leftarrow B$ je také pravidlo, že $P(q_j \leftarrow B) > 0$. Potom *výpočtový krok* z dotazu $? - q_0, q_1, \ldots, q_n$ s použitím pravidla $q_j \leftarrow \boldsymbol{B}$ dáva nový dotaz

$$? - q_0, \ldots, q_{j-1}, C_{\rightarrow}(\boldsymbol{B}, P(q_j \leftarrow B)), q_{j+1}, \ldots, q_n$$

kde $\boldsymbol{B}$ (telo) je výrok tvaru $@(r_1, \ldots, r_k)$, kde $@$ je t-norm alebo t-conorm monotónny agregačný operátor.

Definícia (Vypočítaná odpoveď): Číslo $y \in [0,1]$ sa nazýva *vypočítaná odpoveď* na dotaz q vzhľadom k fuzzy logickému programu P, ak existuje postupnosť $q_0, q_1, \ldots, q_n$ (tiež odpoveď) taká, že

1 T-norm bolo definované v kapitole fuzzy logika

- $q_0 = q$ ($q \in PF$)
- $q_i \rightarrow q_{i+1}$ je výpočtový krok
- q_n neobsahuje žiadne prvotne formule
- y je hodnota výrazu q_n

Príklad programu v tvare FLP by vyzeral teda formúl s príslušnou mierou dôveryhodnosti cf (condence factor):

$$\text{muž}(\text{jozef}).cf = 1$$

$$\text{rodič}(\text{jozef, peter}).\text{cf} = 0.7$$

$$\text{otec}(X, Y) \leftarrow_L \text{muž}(X) \wedge_Z \text{rodič}(X, Y).\text{cf} = 0.9$$

Kde jazyk L je zložený z {muž, rodič, otec, peter, jozef}. Indexy L a Z predstavujú typ použitej spojky fuzzy logiky. Aj keď dôveryhodnosť tretej formule je 0.9 po dosadení faktov namiesto premenných a dopočítaní bude dôveryhodnosť informácie 0,6.

4.5 Bayesove logické programy

Tento model užívateľských preferencií je založený na logickom programovaní a *Bayesových sietiach*. Hlavnou myšlienkou tohto modelu navrhnúť spôsob reprezentácie Bayesových sietí pomocou modifikácie Hornových klauzúl (teda klauzúl logického programovania). Samotná Bayesova sieť obsahuje množinu vrcholov tvoriacich neorientovaný acyklický graf, kde pre každý z týchto vrcholov je daná tabuľka pre podmienené pravdepodobnostné rozdelenie.

Podmienené pravdepodobnostné rozdelenie je reprezentované maticou nazývanou podmienená pravdepodobnostná tabuľka (*conditional probability table* - *CPT*)

Definícia (Bayesova klauzula): *Bayesova klauzula c* je výraz tvaru $A \mid A_1, \ldots, A_n$, kde $n \geq 0$ a $A, A_1, \ldots, A_n$ sú Bayesove atómy[1]. Keď je $n = 0$, potom sa c nazýva *Bayesov fakt* vyjadrený ako A. Takže rozdiel medzi Bayesovými a logickými klauzulami sú:

1. atómy $p(t_1, \ldots, t_l)$ a predikáty p/l predikátovej logiky vytvárajú Bayesove atómy a predikáty čo znamená, že sú spojené s konečnou množinou $S(p/l)$ možných stavov. Inak tiež Bayesovým atómom je priradená konečna množina stavov narozdiel od atómov predikátovej logiky, ktoré majú binárnu hodnotu. Takže je možné predpokladať, že množina všetkých atómov je diskrétna.

1 Bayesove atómy sú atómy, ktoré majú implicitne univerzálny kvantifikátor

2. namiesto znaku „:-“ je použitý „|“ aby sa zdôraznilo podmienené pravdepodobnostné rozdelenie

Bayesova klauzule je zložená, podobne ako Hornova klauzula, hlavu (head) a telo (body). Podmienené pravdepodobnostné rozdelenie priraďuje hlave Bayesovej klauzule *c* pravdepodobnostnú hodnotu v závislosti na tele klauzule *P*(*head*(*c*) | *body*(*c*)).

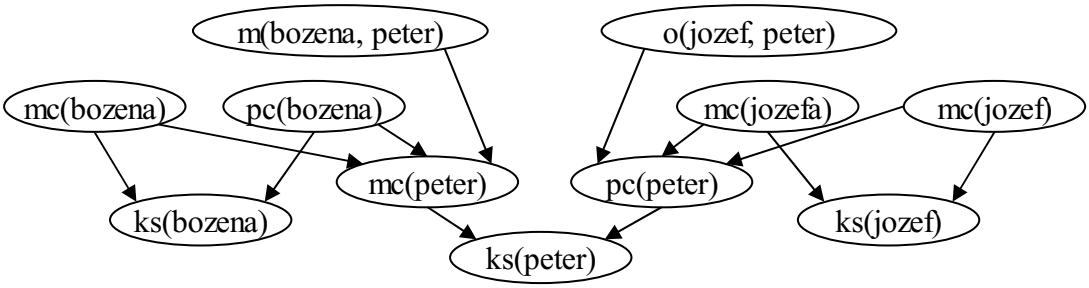
Definícia (Zlučovacie pravidlo): *Zlučovacie pravidlo* (*combining rule - cr*) je funkcia, ktorá mapuje konečnú množinu podmieneného pravdepodobnostného rozdelenia $\{P(A|A_{i1},...,A_{in_i}) \mid i=1,...,m\}$ do jedného podmieneného pravdepodobnostného rozdelenia $P(A \mid B_1, \ldots, B_k)$, kde $\{B_1,...,B_k\} \subseteq \cup_{i=1}^{m} \{A_{i1},...,A_{in_i}\}$.

Inak zlučovacie pravidlo priradí množine Bayesových klauzúl s rovnakou hlavou, ale rozdielnym telom pravdepodobnostnú hodnotu a to v závislosti na rozdielnych pravdepodobnostných rozdelení jednotlivých Bayesových klauzúl. Zlučovacie pravidlo je funkcia ako napríklad maximum, minimum, priemer alebo noisy-or

$$P = 1 - \prod_{\text{pravdivé a možné klauzule } i} (1 - P_{\text{v pripade pravdy}}(i)) \quad .$$

Definícia (Bayesov logický program): *Bayesov logický program B* sa skladá z konečnej množiny Bayesových klauzúl. Pre každú Bayesovu klauzulu *c* obsahuje práve jedno podmienené pravdepodobnostné rozdelenie cpd(*c*) a, pre každý Bayesov predikát *p*/*l* obsahuje práve jedno zlučovanie pravidlo cr(*p*/*l*).

Príklad BLP bude ukázaný na krvných skupinách (tento príklad je prevzatý z [] [Friedman 1999]). Z medicíny je známe, že každý človek má dve kópie chromozómov p-chromozóm (ďalej pc) zdedený po otcovi a m-chromozóm (ďalej mc) zdedený po matke. Každý z týchto chromozómov môže mať hodnotu z množiny {a, b, 0}. Krvná skupina človeka vznikne na základe kombinácie týchto dvoch chromozómov (pc a mc) a môže mať hodnotu {a, b, ab, 0}. Nasledujúci graf ukazuje závislosť tohto príkladu medzi troma osobami otec (jozef), matka (bozena) a dieťa (peter).



Obrázok 4.2, Bayesova sieť príkladu krvných skupín neobsahujúca žiadne premenné

Tento graf krvných skupín je možné reprezentovať pomocou logické programu ako

o(jozef, peter).
m(bozena, peter).
pc(jozef).
mc(jozef).
pc(bozena).
mc(bozena).
pc(peter) :- mc(jozef), pc(jozef).
mc(peter) :- mc(bozena), pc(bozena).
ks(peter) :- mc(peter), pc(peter).
ks(bozena) :- mc(bozena), pc(bozena).
ks(jozef) :- mc(jozef), pc(jozef).

Ak pre každý vrchol grafu bude pridaná podmienená pravdepodobnostná tabuľka vznikne Bayesova sieť. Zovšeobecnením logického programu (pridaním premenných namiesto konštánt), jeho zapísaním do tvaru BLP a pridaním tabuliek podmienenej pravdepodobnostnej distribúcie pre jednotlivé vrcholy a klauzule by Bayesov logicky program vyzeral ako

o(jozef, peter).
m(bozena, peter).
pc(jozef).
mc(jozef).
pc(bozena).
mc(bozena).

mc(X)	pc(X)	P(ks(X))
a	a	(0.97, 0.01, 0.01, 0,01)
a	b	(0.01, 0.01, 0.97 , 0,01)
…	…	…
0	0	(0.01, 0.01, 0.01, 0,97)

pc(X) | o(Y, X), mc(Y), pc(Y).
mc(X) | m(Y, X), mc(Y), pc(Y).
ks(X) | mc(X), pc(X).

m(Y, X)	mc(Y)	pc(Y)	P(mc(X))
true	a	a	(0.98, 0.01, 0.01)
true	b	b	(0.01, 0.98, 0.01)
…	…	…	…
false	a	a	(0.33, 0.33, 0.33)
…	…	…	…

Definícia (Pravdepodobnostný dotaz): *Pravdepodobnostný dotaz* v Bayesovom logickom programe *B* je výraz v tvare:

$$?\text{- } q_1, \ldots, q_n \mid e_1 = e_1, \ldots, e_m = e_m$$

kde $n > 0, m \geq 0$. Je to dotaz na podmienené pravdepodobnostné rozdelenie

$$P(q_1, \ldots, q_n \mid e_1 = e_1, \ldots, e_m = e_m)$$

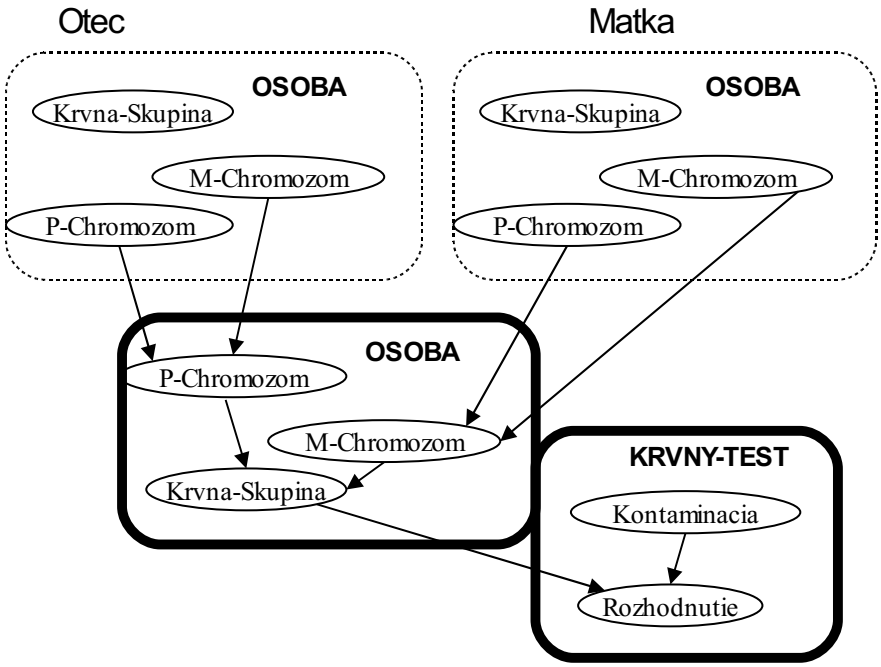
premenných dotazu $q_1, \ldots, q_n$, kde $q_1, \ldots, q_n, e_1, \ldots, e_m \subseteq HB(B)$.

Z vlastností Bayesovej siete intuitívne vyplýva, že k zisteniu pravdepodobnosti nejakého dotazu nie je nutné počítať celý najmenší Herbrandov model. Napríklad k zisteniu krvnej skupiny ks(jozef) je krvná skupina matky nezaujímavá. Preto sa hovorí o tvz. *podpornej sieti* N náhodnej premennej $x \in LH(B)$. N je definovaná ako indukovaná podsieť $\{x\} \cup \{y \mid y \in LH(B) \text{ a } y \text{ ovplivňuje } x\}$. Podporná sieť konečnej množiny $\{x_1, \ldots, x_n\} \subseteq LH(B)$ je zjednotenie všetkých sietí, každého jednotlivého x_i . $LH(B)$ je najmenší Herbrandov model pre Bayesov logický program B. Podporná sieť predstavuje zmenšenie celej Bayesovej siete a znižuje počet nutných výpočtov k zisteniu pravdepodobnosti.

4.6 Pravdepodobnostný relačný model

Tento model vychádza z relačného modelu používaného v databázach (Entity-Relation diagram) a teórie pravdepodobnosti. *Schéma S* relačného modelu obsahuje, množinu *tried* $X_1, \ldots, X_n$. Každá trieda je spojená s množinou *deskriptívnych atribútov* $A(X_i)$ a množinu *relácií* (tiež *reference slot*) $R_1, \ldots, R_m$. Tieto atribúty a relácie, predstavujú priame mapovanie z notácie tried do notácie tabuliek v relačných databázach. Deskriptívne atribúty predstavujú štandardné atribúty tabuľky a relácie predstavujú vzdialené kľúče (kľúčový atribút inej tabuľky). Atribút A triedy X bude zapisovaný ako $A.X$ a relácie triedy X ako $R(X)$. A nakoniec, každý atribút $A_j \in A(X_i)$ nadobúda pevnú doménu hodnôt $V(A_j)$ a bude zapisovaný ako $V(X.A)$.

Model bude ukázaný na rozšírenom príklade krvných skupín, ktorého zadanie bolo popísané pri modele BLP (Bayesove logické programy). Tento príklad bude obsahovať dve triedy a to OSOBA a KRVNY-TEST. Ďalej tri relácie OTEC, MATKA a TEST. Atribúty triedy OSOBA sú Krvná-Skupina, P-Chromozóm a M-Chromozóm. A nakoniec atribúty triedy Krvný-Test budú Dátum, Kontaminácia a Rozhodnutie. P-Chromozóm podobne ako M-Chromozóm môže nadobúdať hodnoty {a, b, 0}, Krvná-Skupina {a, b, ab, 0}. Príklad neobsahuje deterministické atribúty ako Meno, Pohlavie … ktoré sú pre ukážku pravdepodobnostného modelu zbytočné. Popísaný príklad ukazuje nasledujúci obrázok.



Obrázok 4.3, Príklad štruktúra PRM pre krvné skupiny

Na druhej strane pravdepodobnostný model obsahuje štruktúru závislostí *S* a parametre s ňou spojené. Štruktúra závislostí je definovaná, množinou rodičov parents(*X.A*), pre každý atribút *X.A.* To súvisí s *formálnymi rodičmi*, čo zahŕňa aj rodičov *X.A* v tej istej triede *X*. V príklade krvných skupín to sú:

parents(Osoba.M-Chromozóm)=

{Osoba.Matka.M-Chromozóm, Osoba.Matka.P-Chromozóm}

parents(Osoba.P-Chromozóm)=

{Osoba.Otec.M-Chromozóm, Osoba.Otec.P-Chromozóm}

parents(Osoba.Krvná-Skupina)=

{Osoba.M-Chromozóm, Osoba.P-Chromozóm}

parents(Krvný-Test.Rozhodnutie)=

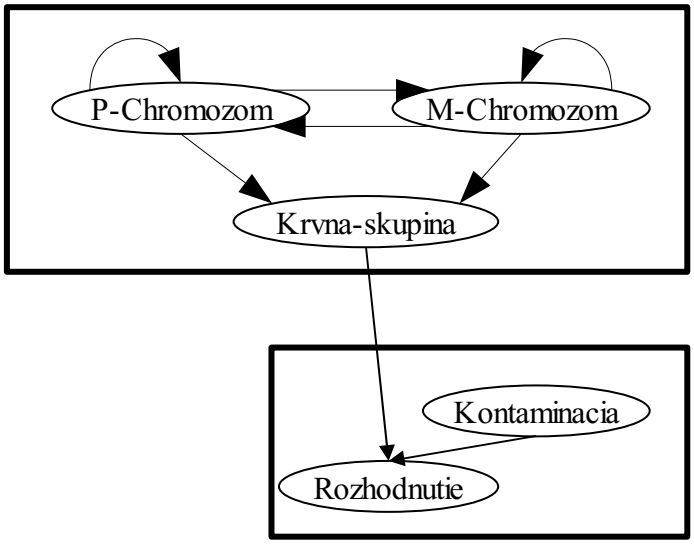
{Krvný-Test.Kontaminácia, Krvný-Test.Osoba.Krvná-Skupina}

Atribút *X.A* môže závisieť na inom podmienenom atribúte *B* tej istej triedy *X*, čo ukazuje aj príklad krvných skupín. Tiež môže závisieť na množine objektov jednej triedy (nie nutne tej istej), ako napríklad priemerná známka z nejakého kurzu, kde nie je úplne jasne koľko jednotlivých hodnotení búdu mať jednotlivý študenti (je to rozdielne). V tomto prípade je možné použiť agregačnu funkciu prípadne inú (min, max, ...) funkciu. Množina rodičov atribútu *A* predstavuje je spojená z podmieneným pravdepodobnostným rozdelením.

Definícia (Probabilistic Relational Model): Probabilistic Relational Model obsahuje:

1. Relačnú schému
2. Špecifikáciu rodičov, každého deskriptívneho atribútu (v tvare cesty)
3. Podmienené pravdepodobnostné rozdelenie, pre každý atribút, každej triedy, ktoré je reprezentované podmienenou pravdepodobnostnou tabuľkou $P(X.A \mid \text{parents}(X.A))$ alebo formou agregačnej funkcie k určeniu pravdepodobností.

Hlavnou myšlienkou PRM je, že všetky informácie jednej entity (respektíve triedy) sú obsiahnuté v jednej relácii.



Obrázok 4.4, Obecné relačné schéma pre príklad krvných skupín rozdelené do tried

V príklade genetiky pre entitu osoba to je osoba(Osoba, PC, MC, KS). Tým sa zamedzuje cyklom v relačnej schéme čo ukazuje predchádzajúci obrázok. Preto je možné PRM prakticky simulovať na databázami, avšak za predpokladu, že budú rozšírené o možnosť pridania podmienenej pravdepodobnostnej tabuľky. V prípade, že relačný model obsahuje cyklus medzi triedami (nezáleží na tom či tie atribúty sú rozdielne) je nutné túto triedu rozdeliť.

4.7 Markovove Logické Siete

Množina formúl v predikátovej logike, ktorá popisuje nejaký problém je pomerne obmedzená. Ak nejaký dotaz porušuje čo i len jednu z formúl jej výsledkom je, že daný dotaz je nepravdivý, čiže má nulovú pravdepodobnosť. Základnou myšlienkou MLN je zjemnenie tohto obmedzenia. To znamená ak nejaký dotaz porušuje nejakú z formúl jeho pravdepodobnosť bude nižšia ale nie nemožná. Teda čím menej formúl porušuje dotaz tým

je pravdepodobnosť väčšia. Každá formula má priradenú váhu, ktorá odráža silu obmedzenia. Inými slovami čím väčšia je váha tým je pravdepodobnosť v prípade splnenia formule väčšia. A naopak pri nesplnení formule s malou váhou bude rozdiel pravdepodobnosti, inak rovnakých dotazov menši.

Definícia (Markov Logic Network): *Markov logic network L* je množina dvojíc (F_i, w_i), kde F_i je formula v predikátovej logike a w_i je reálne číslo. Spolu s konečnou množinou konštánt $C = \{c_1, c_2, \ldots, c_{|C|}\}$ definuje Markovovu sieť $M_{L,C}$ nasledovne:

1. $M_{L,C}$ obsahuje jeden binárny vrchol, pre každý možný predikát objavujúci v L, ktorý neobsahuje premenné. Hodnota vrcholu je 1, ak je pravdivý inak 0.
2. $M_{L,C}$ obsahuje jednu funkciu, pre každú formulu F_i z L, ktorá neobsahuje žiadne premenné. Hodnota týchto funkcií je 1, ak je formula bez premenných pravdivá inak 0. Váha funkcie je w_i združená s F_i v L.

Syntax formúl je štandardná syntax predikátovej logiky. Voľné premenné sú upravené na univerzálny kvantifikátor okrajovú časť formule. Z pohľadu MLN predstavuje šablónu respektíve postup pre konštrukciu Makovových sietí. Pre rozdielnu množinu konštánt, bude vytvárať rozdielnu sieť, čo môže viesť k veľmi meniacej sa veľkosti, ale všetky budú mať určitú pravidelnosť v štruktúre a premenných vychádzajúcej z MLN. To vyplýva z toho, že všetky formule neobsahujúce premenné majú rovnakú váhu. Každá z týchto sietí bude nazývaná *základná (ground) Markovova sieť*. Z definície MLN a vzorcov pre Markovove siete (vid. Kapitola 2.6) je pravdepodobnostné rozdelenie nad Herbrandovou interpretáciou x špecifikované základnou Markovovou sieťou $M_{L,C}$ vzorcom:

$$P(X=x) = \frac{1}{Z}\exp\left(\sum_i w_i n_i(x)\right) = \prod_i \varphi_i(x_{\{i\}})^{n_i(x)}$$

kde $n_i(x)$ je počet pravdivých možností takej formule F_i z x, ktorá neobsahuje žiadne premenné, $x_{\{i\}}$ je stav (pravdivostná hodnota) atómov obsiahnutých v F_i a $\varphi_i(x_{\{i\}}) = e^{w_i}$

Grafická štruktúra vychádzajúca z definície MLN obsahuje hranu medzi dvoma vrcholmi $M_{L,C}$, ak príslušné atómy neobsahujúce premenné, a obsiahnuté prinajmenšom v jednej možnosti aspoň jednej formule (neobsahujúcej premenné) z L. Teda atómy každej formule neobsahujúcej premenné formujú (nie nutne maximálne) kliky v $M_{L,C}$.

Príklad ako vytvoriť Markovovu logickú sieť z dvoch formúl predikátovej logiky a to z výroku „Fajčenie spôsobuje rakovinu.“ a „Ak sú dvaja ľudia priatelia potom fajčia obaja alebo ani jeden.“ zapísané formou predikátovej logiky budú mať výroky tvar

(Smoke, Cancer a Friend).

$$\forall x\, Sm(x) \Rightarrow Ca(x) \quad \text{a} \quad \forall x \forall y \; Fr(x, y) \Rightarrow (Sm(x) \Leftrightarrow Sm(y))$$

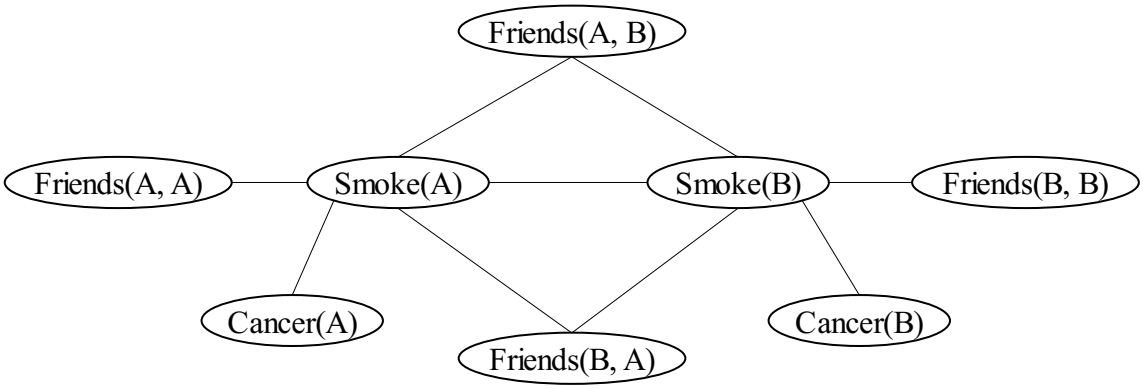
Nakoniec budú prepísané do formy klauzúl (váhy jednotlivých formúl sú brané náhodne).

$\neg Sm(x) \vee Ca(x) \quad 1.5$,

$\neg Fr(x, y) \vee \neg Sm(x) \vee Sm(y) \quad 1.1$

$\neg Fr(x, y) \vee Sm(x) \vee \neg Sm(y) \quad 1.1$

Markovovu logickú sieť pre dvoch priateľov Adama a Boba ukazuje nasledujúci obrázok.



Obrázok 4.5, Štruktúra Markovovej siete vybudovanej z dvoch formúl a dvoch konštánt

Grafu je zložený len z atómov neobsahujúcich premenné ale len konštanty (Adam a Bob). Ďalej obsahuje oblúk medzi každým párom atómov, ktoré sú spolu obsiahnuté v nejakej možností aspoň jednej formule (formule bez premenných). Túto $M_{L,C}$ je teraz možné použiť k odvodeniu pravdepodobnosti, že Adam a Bob sú priatelia na základe toho, že obaja fajčia, alebo pravdepodobnosť, že Bob má rakovinu na základe priateľstva Adamom...

5 Porovnanie modelov

V tejto kapitole budú porovnané jednotlivé modely predstavené v predchádzajúcej kapitole. Budú navrhnuté možné transformácie a prevody medzi nimi a taktiež rozdielna sila, výhody a nevýhody. Nakoniec bude ukázaný graf obsahujúci všetky modely spolu s existujúcimi transformáciami.

5.1 Transformácia do MLN

MLN pracuje s formulami predikátovej logiky, takže teoretický je v ňom možné popísať takmer akúkoľvek situáciu z reálneho sveta (je v ňom možné formulovať širšiu množinu formúl). Tieto situácie nemusia byť nutne úplne pravdivé, ich pravdivosť (respektíve dôležitosť) je určená váhou formule. Nakoniec Markovove site narozdiel od Bayesových sietí nie sú obmedzene acyklickým orientovaným grafom. Ak teda zoberiem príklad s dvoma priateľmi, ktorý fajčia, konkrétne jednu jeho časť a to výrok „Ak sú dvaja ľudia priatelia potom fajčia obaja alebo ani jeden.“ zapísaný formou predikátovej logiky

$$\forall x \forall y \;\; Fr(x, y) \Rightarrow (Sm(x) \Leftrightarrow Sm(y))$$

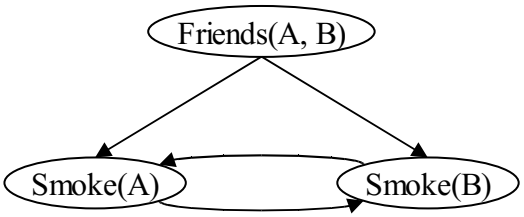
Prepísané do tvaru Hornových klauzúl potrebných k vytvoreniu Bayesovej siete vznikne

$$\neg Fr(x, y) \vee \neg Sm(x) \vee Sm(y) \quad \text{a} \quad \neg Fr(x, y) \vee Sm(x) \vee \neg Sm(y)$$

(váhy v tomto prípade nie sú dôležité) a nakoniec prepísané do tvaru formúl s implikáciou vznikne

$$Fr(x, y) \wedge Sm(x) \rightarrow Sm(y) \quad \text{a} \quad Fr(x, y) \wedge Sm(y) \rightarrow Sm(x) \quad .$$

Graf vytvorený z týchto dvoch formúl by potom vyzeral



Obrázok 5.1, Orientovaný graf formule „Ak sú dvaja ľudia priatelia potom fajčia obaja alebo ani jeden.“

Tento graf však obsahuje cyklus medzi vrcholmi Smoke(A) a Smoke(B), čo Bayesove siete nepripúšťajú. Preto nie je možná úplná transformácia nejakého MLN programu do tvaru

BLP programu. Podobne to platí aj pre PRM, ktorý síce pripúšťa definovanie cyklu medzi atribútmi alebo objektmi jednej triedy, ale zároveň predpokladá, že cyklus nenastane pre rozdielne triedy.

5.1.1 Prevod BLP do MLN

BLP je možné prepísať do MLN vytvorením množiny *D*(...) všetkých predikátov z BLP. Každá formula v BLP je potom obsahuje podmnožinu týchto predikátov a jeden literál predstavujúci rodiča, tento literál taktiež patri do množiny *D*(...). To vyplýva z definície Hornových klauzúl používaných v logickom programovaní. Podmnožina týchto literálov je negovaná, takže tam je jedna formula, pre každú možnú kombináciu pozitívnych a negatívnych literálov. Váha formuly je potom

$$w = \log\left(\frac{p}{1-p}\right)$$

kde p podmienená pravdepodobnosť dieťaťa, keď odpovedajúce spojenie rodičovských literálov je pravda v závislosti na tom aká kombinačná funkcia bola použitá. Ak je kombinačná funkcia logický úsudok môže byť reprezentovaná použitím len lineárnnych čísel formúl.

Pre konkrétny príklad krvných skupín obsahujúci pre predikát m-chromozóm (mc) a p-chromozóm (pc) množinu stavov {a, b, 0} a pre predikát krvná skupina (ks) množinu stavov {a, b, ab, 0}. By model MLN musel vrchol pre konkrétnych troch ľudí otec(jozef), matka(bozena) a dieťa(peter) obsahovať každú kombináciu konštanty a stavu. Takže množina všetkých predikátov konštantu peter by obsahovala

mc(peter_a).	mc(peter_b).	mc(peter_0).	
pc(peter_a).	pc(peter_b).	pc(peter_0).	
ks(peter_a).	ks(peter_b).	ks(peter_0).	ks(peter_ab).

Obdobne tak pre konštanty jozef a bozena. Táto množina by tvorila vrcholy MLN. Ďalej by sa v každej formule BLP postupne nahradzovali premenná za všetky možné predikáty. Tým vznikne množina formúl obsahujúca každú možnosť z podmienených pravdepodobnostných tabuliek. Z tejto podmienenej pravdepodobnosti sa nakoniec logaritmom urči váha jednotlivých formúl.

5.1.2 Prevod PRM do MLN

PRM je možné prepísať do MLN definovaním predikátov $S(x, v)$ pre každý atribút, každej triedy, kde $S(x, v)$ znamená „Hodnota atribútu S v objekte x je v“. PRM je potom preložené do MLN výpisom formúl pre každý riadok každej podmienenej pravdepodobnostnej tabuľky a pre hodnoty atribútov deti. Formula je potom spojenie literálov uvádzajúcich hodnoty rodičov a literálu uvádzajúceho hodnotu dieťaťa, jej váha je logaritmus z $P(x \mid Parents(x))$ odpovedajúcej hodnote v podmienenej pravdepodobnostnej tabuľke (*cpt*). Okrem toho MLN obsahuje formule s nekonečnou váhou uvádzajúcou, že každý atribút musí mať práve jednu hodnotu. Tento prístup zachytáva všetky typy neistoty v PRM.

K ukážke bude opäť použitý príklad krvných skupín. Takže hodnoty atribútov MC, PC a KS v objekte peter (triede osoba) sú

MC(peter, a).	MC(peter, b).	MC(peter, 0)	
PC(peter, a).	PC(peter, b)	PC(peter, 0)	
KS(peter, a).	KS(peter, b).	KS(peter, 0).	KS(peter, ab)

Postup transformácie a tvorba siete je ďalej obdobná ako v predchádzajúcom príklade transformácie BLP do MLN.

5.2 Porovnanie PRM a BLP

PRM je intuitívne rovnaké ako BLP oboch prípadoch je základom acyklický orientovaný graf Bayesových sietí a nim definované podmienene pravdepodobnostné rozdelenie. PRM k tomu využíva relačný model z databáz a BLP rozširuje myšlienku logického programovania. Obecná transformácia medzi nimi je pomerne intuitívna, keďže obe modely boli ukázané na rovnakom príklade. Avšak výhody PRM od BLP sú:

1. umožňuje namiesto podmienenej pravdepodobnostnej tabuľky využiť aj inú možno spojenia atribútov ako napríklad agregačná funkcia.
2. je vhodnejší k samotnej reprezentácii modelu
3. umožňuje rýchlejšie testovať acyklické závislosti
4. definícia PRM vyžaduje konečnú množinu objektov (obdobne aj BLP množinu predikátov) ale pre veľkú množinu objektov by ekvivalentná Bayesova sieť bola veľmi rozsiahla a nepraktická.

K bodu 4 pre konkrétny príklad krvných skupín, v ktorom by figurovalo veľké množstvo ľudí, do veľkej hĺbky by PRM obsahoval jednu tabuľku s obrovským množstvom objektov a reláciami medzi atribútmi. Na druhej strane Bayesova sieť by obsahovala veľkú štruktúru. Takže pre teoreticky nekonečné množstvo objektov je BLP nepoužiteľné.

5.3 Porovnanie FLP a BLP

V článku [16] bola predstavená ekvivalencia medzi modelom FLP a GAP. Ďalej v práci [35] bol ukázaný spôsob transformácie z GAP do BLP. Hlavnou myšlienkou je, že každý anotovaný atóm (je atóm, ktorý má naviac priradenú hodnotu z intervale [0, 1]) bude braný ako jeden predikát. Teda z pohľadu BLP, každý anotovaný atóm predstavuje jeden vrchol Bayesovej siete popisujúcej BLP. Problémom však ostáva, že interval [0, 1] je nekonečný, takže z jedného atómu s rozdielnou anotáciou môže vzniknúť nekonečne veľa rozdielnych vrcholov. Jednou z možností ako sa tomu vyhnúť (teda aj riešiť) je rozdelenie intervalu na niekoľko menších časti (napríklad interval [0.25, 0.75]) s tým, že všetky rovnaké atómy s anotáciou patriacou do toho isté intervalu by reprezentovali jeden vrchol. Ak keď teda existuje formálny prevod z GAP (a teda aj FLP) do BLP pri prevode sa môže stratiť veľká časť informácie (a teda aj sily), ktorú GAP obsahuje.

5.4 Zhrnutie

Boli ukázané jednotlivé prevody medzi modelmi ich výhody a nevýhody, z ktorý ako najvšeobecnejší model je možné považovať Markovove Logické Siete, keďže ponúka najširšiu možnosť k reprezentácií reálneho sveta. Na druhej strane, keďže sú založené na hľadaní klík v grafe, čo je NP-úplný problém to nemusí byť najlepším riešením. Ďalšou nevýhodou je, že navrhnuté transformácie z PRM a BLP do MLN vytvárajú aj pri tak jednoduchom a triviálnom príklade krvných skupín pomerne veľkú Markovovu sieť, ktorá už neposkytuje intuitívne výsledky.

Bayesove Logické Programovanie je pomerne dostatočne silným nástrojom k reprezentácii užívateľských preferencií. Teória BLP je dobre definovaná, samotná Bayesova sieť je pomerne intuitívna a ľahko čitateľná. Nevýhodou však je (čo vyplýva z definície Bayesovej siete) nepripustenie cyklu v grafe a možnosť použitia agregačných funkcií. Tie na druhej strane model PRM dovoľuje, čo ho stavia do výhodnejšej a silnejšej pozície.

6 Príklad preferencií

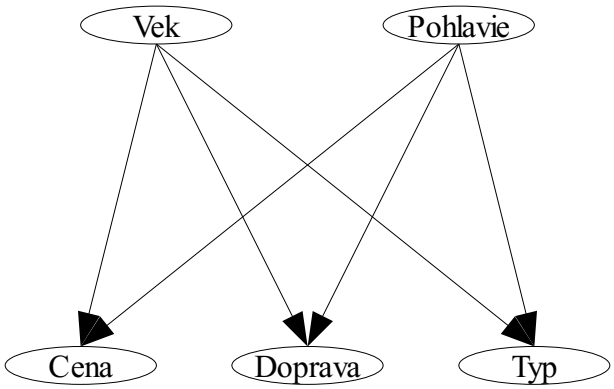
V tejto kapitole bude predstavený príklad na to ako využiť jednotlivé modely k určeniu najvhodnejších preferencií. Teda ako prepojiť samotné preferenciu a dáta. Samotný príklad pozostáva z množiny nejakých vlastností užívateľa a množiny preferencií.

6.1 Zadanie príkladu

Pre prehľadnosť bola množina vlastností o užívateľovi zredukovaná na Vek a Pohlavie. Množina preferencií bola zredukovaná na Cenu zájazdu, Doprava (lietadlo, auto, autobus, loď) a Typ zájazdu (lyžiarske stredisko, turistika, poznávací zájazd, . . .). Ďalej je nutné poznamenať, že na samotná štruktúru môže byť určená k:

- Nájdenie vhodného zájazdu pre užívateľa na základe informácii o ňom
- Určenie vlastností skupiny ľudí pre konkrétny zájazd

Keďže je tento príklad zameraný v istom slova zmysle na druhu predpovedí, teda o hľadaní najvhodnejšieho zájazdu, podľa toho bola navrhnutá vhodná a jednoduchá štruktúra závislostí.



Obrázok 6.1, Jednoduchá štruktúra závislostí medzi turistom a zájazdmi

Zadanie toho príkladu ďalej predpokladá množinu pozorovaní. Teda dostatočne veľkú skupinu ľudí, ktorý si už nejaký konkrétny zájazd vybrali. Na základe týchto informácii je možné určiť tabuľky podmienenej pravdepodobností a tým aj Bayesovu sieť.

Vek			
0-20	21-40	41-60	61-x
0,2	0,3	0,4	0,1

Pohlavie	
Muž	Žena
0,5	0,5

		Cena				
Vek	Pohlavie	0-10000	10k - 20k	20k -30k	30k - 40k	40k - x
0-20	M	0,38	0,25	0,16	0,14	0,07
21-40	M	0,2	0,31	0,24	0,16	0,09
41-60	M	0,12	0,17	0,29	0,22	0,2
61-x	M	0,1	0,25	0,4	0,15	0,1
…	F	…	…	...	...	...

		Doprava		
Vek	Pohlavie	Letecká	Auto	Autobus
0-20	M	0,3	0,5	0,2
21-40	M	0,5	0,2	0,3
41-60	M	0,4	0,1	0,5
61-x	M	0,1	0,5	0,4
…	F	…	…	...

Tabuľky 6.2, Tabuľky podmienenej pravdepodobnosti pre jednotlivé vrcholy predchádzajúcej Bayesovej siete

6.2 Reprezentácia príkladu

Predstavená štruktúra príkladu so zájazdmi je pomerne intuitívna a ľahko pochopiteľná. K jej programovému spracovaniu a spôsobu reprezentácie sú navrhnuté jednotlivé modely predstavené v tomto texte. V stručnosti budú ukázané spôsoby reprezentácie príkladu so zájazdmi pomocou jednotlivých modelov.

6.2.1 BLP tvar

Ako už bolo povedané BLP definuje spôsob reprezentácie Bayesovej siete pomocou logického programu. Na základe navrhnutého príkladu nie je nutné riešiť cyklus v grafe. Zvolená Bayesova sieť by preto v BLP mala tvar.

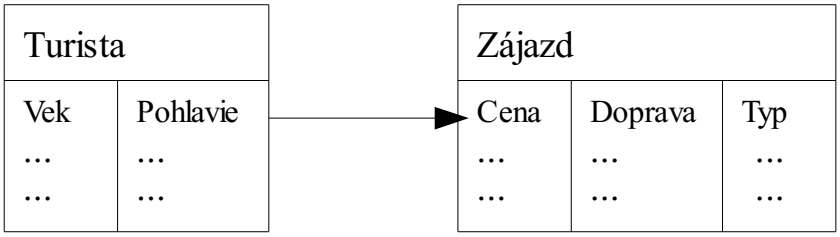
```
vek(X).
pohlavie(X).
doprava(X) | vek(X), pohlavie(X).
typ(X) | vek(X), pohlavie(X).
cena(X) | vek(X), pohlavie(X).
```

Každá z formúl samozrejme musí obsahovať tabuľku podmienenej pravdepodobnosti. A

keďže, každé z pravidiel obsahuje rozdielnu ľavú stranu, nie je nutné žiadne dodatočné kombinačné pravidlo.

6.2.2 PRM tvar

Tento model využíva k reprezentácii Bayesových sietí rozšírenie ER-modelu. Zadaný príklad by v tomto modele bol reprezentovaný závislosťou medzi tabuľkami.



Obrázok 6.3, Bayesova sieť reprezentovaná pomocou PRM modelu

Aj keď sa medzi tabuľkami jedná o prepojenie n:m nie je nutné aby existovala tabuľka obsahujúca ID_Turistu a ID_Zájazdu a taktiež aby tabuľky Turista a Zájazd obsahovali konkrétne ID tejto tabuľky. Jednotlivé závislosti medzi riadkami sú dané tabuľkou podmienenej pravdepodobnosti, pre každý atribút tabuľky. V prípade prepojenia by pre turistu existoval jednoduchý SQL dotaz, ktorého výsledkom by bola množina (aj prázdna) všetkých Zájazdov, ktoré už absolvoval ale nie pravdepodobnosť aký druh zájazdov preferuje skupina ľudí s rovnakými vlastnosťami. Samotné prepojenie by bolo nutné k určeniu štruktúry Bayesovej siete a k získaniu tabuliek podmienenej pravdepodobnostnej distribúcie, teda učenie[1].

6.2.3 MLN tvar

Markovove logické siete reprezentujú formule logiky prvého rádu pomocou Markovových sieti Na základe navrhnutej možnej transformácie a predpokladu MLN, že každý vrchol môže obsahovať len binárne hodnoty by jednotlivé formule mali tvar.

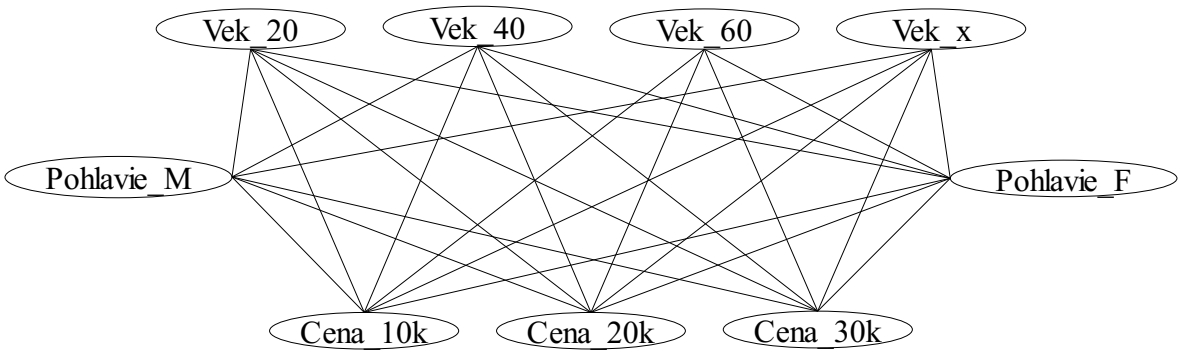
$$\neg Vek_{20}(X) \vee \neg Pohlavie_{M}(X) \vee Cena_{10k}(X) \quad 1.2$$

$$\neg Vek_{20}(X) \vee \neg Pohlavie_{M}(X) \vee Cena_{20k}(X) \quad 1.4$$

$$\neg Vek_{20}(X) \vee \neg Pohlavie_{M}(X) \vee Cena_{30k}(X) \quad 0.9$$

Obdobne by to vyzeralo, pre každú možnú kombináciu.

1 Tento text je zameraný na rozdielne možnosti reprezentácie užívateľských preferencií, nie na možnosti samotného učenia.



Obrázok 6.4, Bayesova sieť príkladu turista-zájazd prevedená do tvaru Markovovej siete

V použitej transformácii tomto príklade nie je výhoda Markovových sieti úplne jasné. Bolo by však jednoduchšie modelovanie vzťahu medzi kombináciou preferencií dvoch (a viac) ľudí, ktorých preferencie pri voľbe zájazdu úzko súvisia. Ako napríklad mladomanželský pár (väčšinou rozdielne pohlavie a nízky vekový rozdiel), rodina (dvaja rovnaký vek ostatný sú mladší) alebo skupina študentov (veľa ľudí približne rovnako starý).

Tieto závislosti je samozrejme možné reprezentovať samotnou Bayesovou sieťou, ale je nutné (z definície Bayesovej siete) dávať pozor na prípadný cyklus v grafe.

6.3 Určenie preferencií

Ako už bolo povedané v kapitole o preferenciách, modelovanie preferencií môže byť založené na základe:

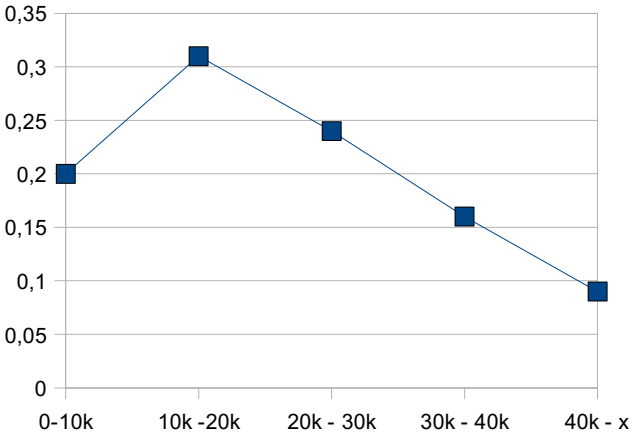
- informáciach o užívateľoch
- ohodnocovaní objektov
- špecifikovaní preferencií

Pre konkrétny príklad turista-zájazd predstavujú informácia ako Vek a Pohlavie informácie o užívateľoch, avšak z pohľadu konkrétneho užívateľa ho len zaraďujú do istej preferenčnej skupiny. Cena, Doprava a Typ predstavujú samotné preferencie. Každý užívateľ môže samozrejme mať rozdielne preferencie, ak však nie sú konkretizované (respektíve užívateľ definoval len niektoré z nich) potom na základe vlastností spadá do pravdepodobnostnej množiny preferencií. Samotné ohodnocovanie objektov nie je zahrnuté, keďže z pohľadu užívateľa sa jedná spätné ohodnotenie (najprv pôjde na zájazd potom môže povedať ako s nim bol spokojný). Samozrejme je možné pridať do grafu vlasnosť (napríklad Rating) spolu so závislosťami v grafe a pravdepodobnostnou tabuľkou alebo brať rating jednotlivých zájazdov ako hodnotu na usporiadanie konečnej skupiny

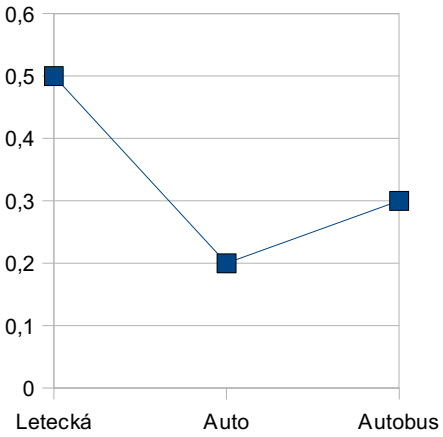
zájazdov, ktorá užívateľovi vyhovuje.

Z databázového pohľadu pohľadu príkladu turista-zájazd je potrebné aby existovala tabuľka obsahujúca zoznam zájazdov s vlastnosťami (Cena, Doprava a Typ). Samotnou úlohou potom je na základe preferencií konkrétneho užívateľa nájsť top-k najviac vyhovujúcich zájazdov. Jednou z možností je použitie Faginovho, ktorý potrebuje, určiť fuzzy funkciu predstavujúcu preferencie pre aspoň jeden z atribútov. Teda určenie fuzzy funkcie pre aspoň jednu vlastnosť z Cena, Doprava alebo Typ.

Ak nie sú známe všetky preferencie potenciálneho zákazníka, ale sú známe jeho vlastnosti (Vek a Pohlavie) je možné určiť fuzzy funkcie pre jednotlivé atribúty na základe tabuľky podmienenej pravdepodobnosti



Obrázok 6.5, Fuzzy funkcia preferencií Ceny pre skupinu ľudí vo veku od 21-40 rokov.



Obrázok 6.6, Fuzzy funkcia preferencií diskrétneho atribútu Doprava pre skupinu ľudí vo veku od 21-40 rokov.

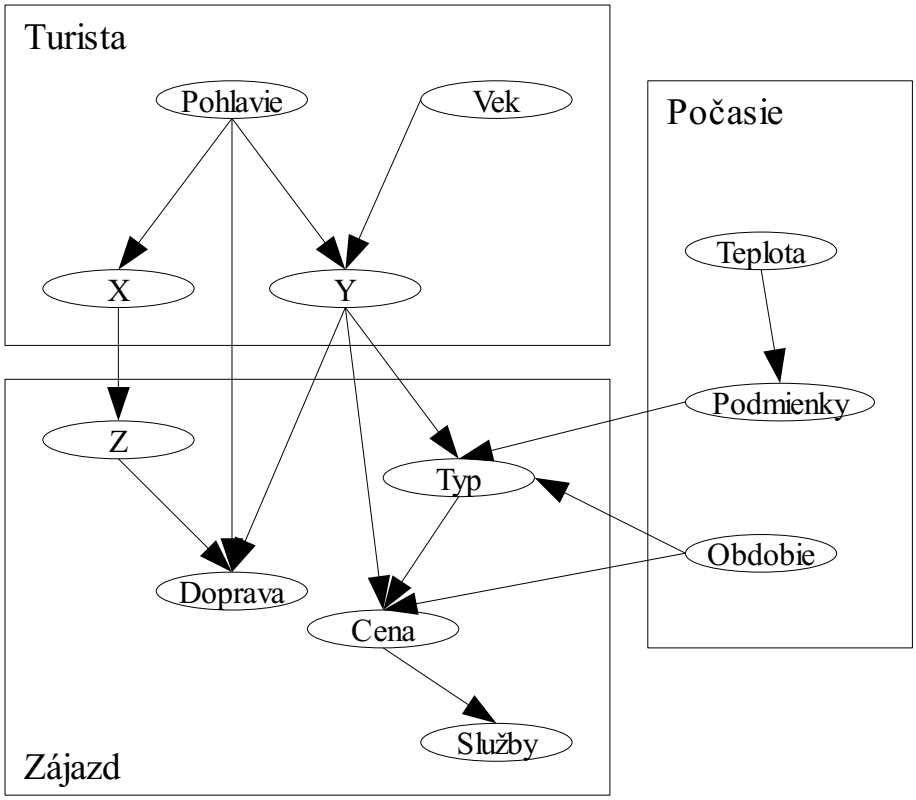
Ak zákazník zadá konkrétnu preferenciu jednoducho bude nahradená pri hľadaní top-k zájazdov namiesto pravdepodobnostnej funkcie. Ak o zákazníkovi nie sú známe všetky

informácie (len Vek alebo Pohlavie) budú všetky možné výsledky zlúčené do jednej pravdepodobnosti.

6.4 Zovšeobecnenie príkladu

Predstavený príklad je samozrejme len motivačný a obsahuje len malú množinu vlastností a preferencií, takže otázkou ostáva aká je výhoda samotného modelu ak samotná tabuľka pravdepodobností nemusí byť vôbec reprezentovaná a v každom dotaze by sa jednoducho dopočítala na základe podobných vlastností. Reprezentovanie preferencií pomocou modelu má niekoľko výhod

- jednou z výhod je, že samotný graf závislostí nemusí byť úplne triviálny kde všetko súvisí zo všetkým ale niektoré závislosti tam nie sú
- druhou je, šetrenie miesta, čo je dosť určujúce pri veľkom množstve navzájom závislých vrcholov
- treťou je rýchla práca v prípade zmeny pravdepodobností v tabuľke na základe posledného rozhodnutia



Obrázok 6.7, Ukážka rozšírenej štruktúry závislosti príkladu turista-zájazd

7 Návrhy na rozšírenie

Aj keď bola ukázaná ekvivalencia alebo možná transformácia medzi jednotlivými modelmi, v niektorých prípadoch sa stratila sila jedného modelu alebo sa neukázala sila iného modelu. Jednou z možnosti ako tieto straty eliminovať je navrhnúť všeobecnejší pravdepodobnostný model, ktorý by zahŕňal možnosť pravdivosti pomocou fuzzy logiky. Rozdielom od stávajúcich modelov by bolo, že pravdepodobnostný model by namiesto klasickej tabuľky pre podmienené pravdepodobnostné rozdelenie obsahoval funkciu k učeniu pravdepodobnosti. Táto funkcia by mohla vychádzať z pozorovania a dlhodobých štatistík.

Z druhej strany, z pohľadu FLP, by bolo možné rozšíriť každé z pravidiel a faktov o ďalšiu hodnotu odpovedajúcu váhe daného faktu alebo pravidla. Tým sa model stal silnejším z dôvodu, že už by neodpovedal len na základné otázky pravdivosti ale umožnil by aj rozhodnutie v prípade dvoch identických (obsahujúcich rovnaké atómy) pravidiel s rozdielnou pravdivosťou. Táto váha by mohla predstavovať aj hodnotu počtu ľudí, ktorý si dané pravidlo zvolili alebo nie, respektíve logaritmus z počtu ľudí.

Ďalšou možnosťou je zahrnutie do modelov rozhodovací strom tak, že každý vrchol rozhodovacieho stromu by obsahoval identickú Bayesovu sieť. Ak by sa zobrali siete, ktoré by simulovali (respektíve vyjadrovali) stavy pacienta v časovom slede. Tieto jednotlivé siete by mohli predstavovať vrcholy v rozhodovacom strome. Tým by bolo možné simulovať dlhodobú diagnózu s menšími pamäťovými a výpočtovými nárokmi, keďže závislosti v Bayesovej siete, ktoré by vychádzali z dlhodobých diagnóz by boli obsiahnuté v rozhodovacom strome.

8 Záver

Cieľom práce bol teoretický rozbor tematiky užívateľského rozhodovania a modelov užívateľských preferencií. Táto oblasť zahŕňa Bayesove siete, viachodnotové logické programy, zovšeobecnené anotované programy a pravdepodobnostné logické programy. Ťažiskom práce sa stali modely založené na teórii Bayesových sietí a Markovových sieti vychádzajúce z teórie pravdepodobnosti.

Práca je rozdelená do niekoľkých časti. V prvej časti boli rozobrané a predstavené rôzne pohľady na užívateľské preferencie obecne. Základným hľadiskom bolo uplatnenie zo strany stávajúceho reálneho sveta. Táto časť ďalej obsahuje prehľad jednotlivých modelov užívateľských preferencií, z ktorých niektoré a boli rozobrané a ukázané na príklade.

V druhej časti boli porovnané jednotlivé modely. A medzi jednotlivými modelmi boli navrhnuté a ukázané možné transformácie. V závere bola zhodnotená sila, výhody a nevýhody jednotlivých modelov.

V poslednej časti bol ukázaný príklad z reálneho sveta a to navrhnutie najvhodnejšieho zájazdu pre potencionálneho zákazníka (turistu). Príklad obsahuje možnosti spojenia preferencií a dát v využitím jednotlivých modelov. V poslednej kapitole boli navrhnuté možnosti na rozšírenia modelov. Tiež boli navrhnuté rôzne kombinácie medzi jednotlivými technikami používanými v modelovaní užívateľských preferencií.

Prínos tejto práce je to, že zahŕňa široký okruh nástrojov, používaných v oblasti modelovania užívateľských preferencií. Jednotlivé kapitoly sú navrhnuté tak, aby na seba naväzovali a to od najjednoduchších a najzákladnejších až po netriviálne a zložitejšie. Preto je vhodná ako doplňujúci materiál pre skriptá k predmetu užívateľské preferencie. Ďalším prínosom je zostavenie širokého prehľadu v pomerne mladej oblastí informatiky v ktorej doteraz ešte nebol zostavený komplexnejší prehľad.

Motiváciou pre ďalšie rozšírenie práce je podrobné rozobratie všetky stávajúce modely, či už tie, ktorých ekvivalencie s predstavenými modelmi boli už ukázané niekým iným, alebo modely vychádzajúce z teórií, ktorých prevody a transformácie nie sú úplne triviálne (Dynamické Bayesove siete). Taktiež sa nezaoberá možnosťami, ktoré síce je možné použiť v preferenčnom modelovaní ale ich aplikáciou sa doposiaľ nikto nezaoberal, ako napríklad shlukovacie metódy a neuronové siete.

9 Literatúra

[1] Cussens J. (1999): *Loglinear models for rst-order probabilistic reasoning*. Proceedings of the Fifteenth Conference on on Uncertainty in Articial Intelligence, 126-133.

[2] Domingos P. & Richardson M. (2006): *Markov logic networks*. Machine Learning 62, 107-136.

[3] Domingos P. & Richardson M. (2007): *Markov Logic: A Unifying Framework for Statistical Relational Learning.* Introduction to Statistical Relational Learning 339-371.

[4] Dvořák T. (2006): I*nduction of user preferences in semantic web*. Charles University, Faculty of Mathematics and Physics, Prague

[5] Friedman N., Getoor L., Koller D. & Pfeffer A. (2001): *Learning probabilistic relational models*. Relational Data Mining

[6] Friedman N., Getoor L., Koller D. & Pfeffer A. (1999): *Learning probabilistic relational models*. Proceedings of the Sixteenth International Joint Conference on Articial Intelligence, 1300-1307.

[7] Getoor L., Koller D., Taskar B. & Friedman N. (2001): *Learning Probabilistic Models of Relational Structure*. Proceedings of the Eighteenth International Conference on Machine Learning, 170-177.

[8] Getoor L., Koller D., Taskar B. & Friedman N. (2000): *Learning probabilistic relational models with structural uncertainty.* Proceedings of the AAAI-2000 Workshop on Learning Statistical Models from Relational Data, 13-20.

[9] Heckerman D., Chickering D. M., Meek C., Rounthwaite R. & Kadie C. (2000): *Dependency networks for inference, collaborative filtering*. Journal of Machine Learning Research, and datavisualization, 49-75.

[10] Heckerman D., Meek C. & Koller D. (2004): *Probabilistic entity-relationship models, PRMs, and plate models*. Proceedings of the ICML-2004 Workshopon Statistical Relational Learning and its Connections to Other Fields, 55-60.

[11] Horváth T. & Vojtáš P. (2004): GAP - *Rule Discovery for Graded Classifciation*. Workshop of Advances in Inductive Rule Learning, 46-63.

[12] Kersting K. & De Raedt L. (2001): *Bayesian Logic Programs*. Technical Report 52.

[13] Kersting K. & De Raedt L. (2001): *Adaptive Bayesian Logic Programs*. Proceedings of the Eleventh Conference on Inductive Logic Programming 2157, 104 – 117.

[14] Kersting K. & De Raedt L. (2001): *Towards combining inductive logic programming with Bayesian networks*. Proceedings of the Eleventh International Conference on Inductive Logic Programming, 118-131.

[15] Kifer M. & Subrahmanian V. S. (1992): *Theory of generalized annotated logic programming and its applications*. Logic Programing, 12, 335-367.

[16] Krajčí S., Lencses R. & Vojtáš P. (2004): *A comparison of fuzzy and annotated logic programming*. Fuzzy sets and systems 144, 173-192.

[17] Zadeh L.A. (1975): *The concept of a linguistic variable and its application to approximate reasoning*. Information Sciences 1, 119-249.

[18] Zadeh L.A. (1965): *Fuzzy sets*. Information Control 8, 338-353.

[19] Muggleton S. (1996): *Stochastic logic programs*. Advances in inductive logic programming, 254-264.

[20] Neville J. & Jensen D. (2003): *Collective classication with relational dependency networks*. Proceedings of the Second International Workshop on Multi-Relational Data Mining, 77-91.

[21] Ng R. and & Subrahmanian V.S. (1992): *Probabilistic Logic Programming*. Information and Computation 101, 2, 150-201.

[22] Ngo L. & Haddawy P. (1997): *Answering queries from context-sensitive probabilistic knowledge bases*. Theoretical Computer Science 171, 147-177.

[23] Poole D. (2003): *First-order probabilistic inference*. Proceedings of the Eighteenth International Joint Conference on Articial Intelligence, 985-991.

[24] Poole D. (1993): *Probabilistic Horn abduction and Bayesian networks*. Articial Intelligence, 81-129.

[25] [25] Puech A. & Muggleton S. H. (2003): *A comparison of stochastic logic programs and Bayesian logic programs*. Workshop on Learning Statistical Models from Relational Data

[26] Rafee Ebrahim (2001): *Fuzzy logic programming*. Fuzzy Sets and Systems 117, 215-230.

[27] Riezler S. (1998): *Probabilistic constraint logic programming*. Doctoral dissertation

[28] Sato T. & Kameya Y. (1997): *PRISM: A symbolic-statistical modeling language*. Proceedings of the Fifteenth International Joint Conference on Articial Intelligence, 1330-1335.

[29] Sato T. and Kameya Y. (2008): *New advances in logic-based probabilistic modeling by PRISM*. Probabilistic Inductive Logic Programming, 118-155.

[30] Sung Young Jung, Jeong-Hee Hong & Taek-Soo Kim (2005): *A statistical model for user preference*. Knowledge and Data Engineering, IEEE Transactions on 17, 6, 834-843.

[31] Taskar B., Abbeel P. & Koller, D. (2002): *Discriminative probabilistic models for relational data*. Proceedings of the Eighteenth Conference on Uncertainty inArticial Intelligence, 485-492.

[32] Taskar B., Abbeel P., Wong M.-F. & Koller D. (2007): *Relational Markov Networks.* Introduction to Statistical Relational Learning, 176-199.

[33] Berners-Lee T., Hendler J. & Ora Lassila (2001): *The Semantic Web*. Scientific American.

[34] Vojtáš P. (2001): *Fuzzy logic programming*. Fuzzy Sets and Systems 124, 3, 361-370.

[35] Vojtáš P. & Vomlelová M. (2006): *On models of comparison of multiple monotone classifications*. Proc. IPMU 2006, 1236-1243.

[36] Wellman M., Breese J. S. & Goldman R. P. (1992): *From knowledge bases to decision models*. Knowledge Engineering Review, 7.

A Obsah CD

Priložený CD disk obsahuje text diplomovej práce vo formáte pdf s názvom

`diplomova_praca_peter_lacky.pdf`.