

**METODY REPREZENTACE
A ZPRACOVÁNÍ ZNALOSTÍ
V UMĚLÉ INTELIGENCI**

Radim Jiroušek

Obsah

0	Úvod	7
1	Řešení úloh ve stavových prostorech	9
1.1	Definice stavového prostoru	9
1.2	Příklad	10
1.3	Optimalita řešení	12
1.4	Metody hledání řešení	13
1.5	Řízené prohledávání grafů	15
1.6	Cvičení	19
1.7	Použitá literatura	20
2	Obecný rozhodovací problém	21
2.1	Rozhodovací tabulky	23
2.2	Rozhodovací stromy	24
2.3	Střední délka rozhodovacích stromů	29
2.4	Metoda větví a mezí	33
2.5	Cvičení	35
2.6	Literatura	36

3	Lineární rozhodovací funkce	37
3.1	Rozhodnutí lineárně separabilní	38
3.2	Příklady	42
3.3	Rozhodnutí lineárně neseperabilní	43
3.4	Poznámka o neuronových sítích	45
3.5	Cvičení	46
3.6	Literatura	48
4	Chyba rozhodování	49
4.1	Přípustné rozhodovací funkce	49
4.2	Optimalita rozhodovacích funkcí	53
4.3	Rozhodování za neurčitosti	55
4.4	Cvičení	58
4.5	Literatura	58
5	Skládání částečných znalostí	59
5.1	Pseudobayesovský přístup	59
5.2	Metoda korekcí	62
5.3	Cvičení	67
5.4	Literatura	68
6	Bayesovské sítě	69
6.1	Distribuce reprezentovaná bayesovskou sítí	70
6.2	Odvozování metodou postupných modifikací	77
6.3	Transformace na rozložitelný model	81

<i>OBSAH</i>	5
6.4 Výpočty ve stromech spojení	85
6.5 Cvičení	90
6.6 Literatura	91
A Základní pojmy konečné teorie pravděpodobnosti	93
A.1 Pravděpodobnostní rozložení	93
A.2 Náhodné veličiny	96
A.3 Pojmy teorie informace	99
A.4 Cvičení	101
A.5 Literatura	101
Rejstřík	102

Kapitola 0

Úvod

V literatuře se lze setkat s mnoha definicemi pojmu *znalost*. Žádná z nich však není natolik dobrá, aby byla všeobecně přijímána. Proto i v tomto učebním textu zvolíme často používaný přístup, který vychází z předpokladu, že znalost je základním pojmem, který je každému jasný a srozumitelný a že jej není třeba definovat.

Při takovémto přístupu je však pojem znalosti přece jenom příliš široký a proto si hned na začátku řekneme, že ne každou znalostí se budeme v tomto textu zabývat. Asi se shodneme na tom, že znalostí rozumíme něco, co nám umožňuje využívat zkušeností (vlastních i cizích), nacházet řešení různých problémů či efektivně se rozhodovat. Jinými slovy, něco, co nám umožňuje chovat se *inteligentně*. Znalosti tohoto druhu je často možno rozdělit do dvou skupin, podle toho, jakým způsobem je sdělujeme, na znalosti *deklarativní* a *procedurální*¹.

K výpočtu kořenů kvadratické rovnice

$$ax^2 + bx + c = 0$$

je jistě třeba určité znalosti. Tuto můžeme mít vyjádřenu pomocí vzorce na výpočet kořenů kvadratické rovnice

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a},$$

či pomocí algoritmu realizujícího metodu největšího spádu, tzv. Newtonovu metodu. První způsob je klasickým příkladem znalosti vyjádřené deklarativním způsobem, způsob druhý odpovídá procedurálnímu vyjádření znalosti. Na tomto příkladu je také vidět některé obecné vlastnosti těchto dvou způsobů vyjádření. Způsob deklarativní je obvykle stručnější, vyžaduje však od toho, kdo jej využívá, některé další znalosti. Ty se

¹I toto dělení je třeba brát velice opatrně, neboť jedna znalost může být sdělena oběma “formami” a často lze těžko tyto formy od sebe jednoznačně oddělit.

týkají jak zvyklostí používaných při vlastním deklarativním záznamu znalostí (v našem případě např. matematické značky vyjadřující zlomek a odmocninu) tak i znalostí obecnějších (např. umět vypočítat druhou odmocninu, vědět, jak se zachovat, když je diskriminant záporný a pod.). Procedurální znalosti však bývají často hůře (či složitěji) sdělitelné.

Existuje však ještě celá řada znalostí, které nelze zařadit ani do jedné z těchto tříd. Jedná se o znalosti *neuvědomělé*. Jako příklad tohoto typu znalostí bývá v literatuře uváděna schopnost chůze. K tomu, abychom chodili, určitě nějakou znalost potřebujeme. Neuvědujeme si ji však a také ji neumíme sdělit ani zaznamenat. Takovýmito znalostmi se v našem textu samozřejmě zabývat nebudeme. Převážně se budeme zabývat znalostmi deklarativními, způsobem jejich zaznamenání a především způsobem jejich efektivního využívání převážně pro rozhodování. Z tohoto rámce trochu vybočuje první kapitola, která se nezabývá rozhodováním, ale řešením problémů. Jedná se o natolik obecný přístup patřící v současné době již mezi klasické metody, že jej nebylo možno v tomto textu opominout.

Kapitola 1

Řešení úloh ve stavových prostorech

Formalizační aparát a techniky popsané v této kapitole patří mezi klasické metody pro reprezentaci a využívání znalostí používané v umělé inteligenci již od samého začátku. Technika prohledávání stavových prostorů pro řešení problémů byla společně s technikou automatizovaného dokazování teorémů vyvinuta již na rozhraní 50. a 60. let. V tomto učebním textu se jí zabýváme nejenom proto, že se jedná vlastně o jednu z prvních metod využívajících znalostí o řešení úloze (viz zavedení heuristické funkce), ale také proto, abychom mohli ukázat, že zde používaný algoritmus \mathbf{A}^* je speciálním případem obecné metody větví a mezí probírané později.

1.1 Definice stavového prostoru

Stavovým prostorem budeme rozumět uspořádanou dvojici $\mathcal{S} = (S, \Phi)$, kde

- S je konečná množina stavů,
- Φ je konečná množina operátorů.

Operátorem φ v tomto kontextu rozumíme parciální zobrazení¹ množiny stavů S do sebe.

Úloha je uspořádaná trojice (\mathcal{S}, s_0, G) , kde

- \mathcal{S} je stavový prostor,
- $s_0 \in S$ je počáteční stav a
- $G \subset S$ je množina cílových stavů.

Za *řešení* úlohy (\mathcal{S}, s_0, G) budeme považovat každou konečnou posloupnost operátorů

¹Parciální zobrazení φ množiny S , na rozdíl od běžného zobrazení (které také bývá nazýváno zobrazením totálním), nemusí být definováno pro všechny prvky množiny S . Je-li pro nějaké $s \in S$ hodnota $\varphi(s)$ definována, potom říkáme, že zobrazení je *aplikovatelné* nebo *použitelné* na s .

$(\varphi_1, \varphi_2, \dots, \varphi_n)$ takovou, že

- operátor φ_1 je použitelný na stav s_0 ,
- operátor φ_2 je použitelný na stav $\varphi_1(s_0)$,
- operátor φ_3 je použitelný na stav $\varphi_2(\varphi_1(s_0))$,
-
-
-
- operátor φ_n je použitelný na stav $\varphi_{n-1}(\dots\varphi_2(\varphi_1(s_0))\dots)$,
- a konečně stav $\varphi_n(\varphi_{n-1}(\dots\varphi_2(\varphi_1(s_0))\dots)) \in G$.

Číslo n budeme nazývat *délkou* řešení $(\varphi_1, \varphi_2, \dots, \varphi_n)$. V případě, že $s_0 \in G$ budeme za řešení nulové délky považovat i prázdnou posloupnost.

1.2 Příklad

Uvažujme následující logickou hříčku. Tři cestovatelé putují zemí lidojedů a mají s sebou tři domorodé průvodce. Příliš jim nedůvěřují a tak se snaží, aby nad nimi neměli domorodci početní převahu. Všichni se potřebují dostat na druhý břeh řeky, ale mají k dispozici jen dvoumístnou loďku.

Při formalizaci této hříčky musíme nejprve navrhnout, co budeme považovat za jednotlivé stavy náležející do množiny S . Zde se přirozeně nabízí popsat situaci nastávající před vyjetím (či po přistání) loďky. Vzhledem k tomu, že v námi uvažovaných stavech je loďka u břehu, můžeme stavy jednoznačně popsat počtem cestovatelů a domorodců na původním břehu. Zápis stavu $[3, 3]$ bude označovat situaci výchozí, kdy všichni tři cestovatelé i domorodci jsou na původním břehu. Podobně zápis $[2, 1]$ bude popisovat nedovolený stav, při které jsou sice dva cestovatelé a jeden domorodec na původním břehu, ale na protějším břehu je jeden cestovatel se dvěma domorodci, což je pravidly hříčky zapovězeno.

Pro hledání řešení úlohy však nestačí vědět, kolik cestovatelů a domorodců je na kterém břehu, ale musíme též vědět, na kterém břehu je loďka. Budeme tedy za dvojici čísel ještě připsávat znaménko $+$ v situacích, kdy je loďka na původním břehu a $-$, když bude loďka u břehu protějšího.

Stavový prostor S tedy obsahuje 18 *povolených* stavů²:

$$\begin{array}{cccccccc}
 [3, 3, +] & & [3, 2, +] & [3, 2, -] & [3, 1, +] & [3, 1, -] & [3, 0, +] & [3, 0, -] \\
 & & [2, 2, +] & [2, 2, -] & [1, 1, +] & [1, 1, -] & & \\
 [0, 3, +] & [0, 3, -] & [0, 2, +] & [0, 2, -] & [0, 1, +] & [0, 1, -] & & [0, 0, -]
 \end{array}$$

Operátorů budeme definovat 5:

$\varphi_{(10)}$ - pluje jeden cestovatel,

$\varphi_{(20)}$ - plují dva cestovatelé,

$\varphi_{(01)}$ - pluje jeden domorodec,

$\varphi_{(02)}$ - plují dva domorodci,

$\varphi_{(11)}$ - pluje jeden cestovatel a jeden domorodec.

Každý z nich je aplikovatelný pouze na některé stavy (tedy všechny jsou parciálními zobrazeními). Definice těchto zobrazení jsou v následujících tabulkách:

$\varphi_{(10)}$	
$[1, 1, +]$	$\longrightarrow [0, 1, -]$
$[0, 1, -]$	$\longrightarrow [1, 1, +]$

$\varphi_{(20)}$	
$[3, 1, +]$	$\longrightarrow [1, 1, -]$
$[1, 1, -]$	$\longrightarrow [3, 1, +]$
$[2, 2, +]$	$\longrightarrow [0, 2, -]$
$[0, 2, -]$	$\longrightarrow [2, 2, +]$

$\varphi_{(11)}$	
$[3, 3, +]$	$\longrightarrow [2, 2, -]$
$[2, 2, -]$	$\longrightarrow [3, 3, +]$
$[2, 2, +]$	$\longrightarrow [1, 1, -]$
$[1, 1, -]$	$\longrightarrow [2, 2, +]$
$[1, 1, +]$	$\longrightarrow [0, 0, -]$
$[0, 0, -]$	$\longrightarrow [1, 1, +]$

²Stavy $[3, 3, -]$ a $[0, 0, +]$ sice nejsou zakázány pravidly, ale nemohou nastat, protože popisují situace, kdy je loďka na jiném břehu než všichni cestovatelé i domorodci.

$\varphi(01)$	
$[3, 3, +]$	$\longrightarrow [3, 2, -]$
$[3, 2, -]$	$\longrightarrow [3, 3, +]$
$[3, 2, +]$	$\longrightarrow [3, 1, -]$
$[3, 1, -]$	$\longrightarrow [3, 2, +]$
$[3, 1, +]$	$\longrightarrow [3, 0, -]$
$[3, 0, -]$	$\longrightarrow [3, 1, +]$
$[0, 3, +]$	$\longrightarrow [0, 2, -]$
$[0, 2, -]$	$\longrightarrow [0, 3, +]$
$[0, 2, +]$	$\longrightarrow [0, 1, -]$
$[0, 1, -]$	$\longrightarrow [0, 2, +]$
$[0, 1, +]$	$\longrightarrow [0, 0, -]$
$[0, 0, -]$	$\longrightarrow [0, 1, +]$

$\varphi(02)$	
$[3, 3, +]$	$\longrightarrow [3, 1, -]$
$[3, 1, -]$	$\longrightarrow [3, 3, +]$
$[3, 2, +]$	$\longrightarrow [3, 0, -]$
$[3, 0, -]$	$\longrightarrow [3, 2, +]$
$[0, 3, +]$	$\longrightarrow [0, 1, -]$
$[0, 1, -]$	$\longrightarrow [0, 3, +]$
$[0, 2, +]$	$\longrightarrow [0, 0, -]$
$[0, 0, -]$	$\longrightarrow [0, 2, +]$

Odpovídající stavový prostor je tedy definován danou množinou 18 stavů a množinou 5 operátorů $\Phi = \{\varphi(10), \varphi(20), \varphi(11), \varphi(01), \varphi(02)\}$. Úloha je pak dána navíc definicí počátečního stavu $s_0 = [3, 3, +]$ a jednoprvkovou množinou cílových stavů $G = \{[0, 0, -]\}$. Ponecháváme na čtenáři, aby ověřil, že posloupnost

$$\varphi(02), \varphi(01), \varphi(02), \varphi(01), \varphi(20), \varphi(11), \varphi(20), \varphi(01), \varphi(02), \varphi(01), \varphi(02)$$

je řešením dané úlohy. Délka tohoto řešení je 11.

1.3 Optimálnost řešení

Při hledání řešení úlohy nás budou samozřejmě zajímat ta řešení, která dosáhnou cílového stavu co nejdříve. Proto se budeme zajímat především o to, zda umíme nalézt řešení *optimální*, neboli řešení, které má nejmenší délku. Délku optimálního řešení úlohy (\mathcal{S}, s_0, G) budeme označovat symbolem $\text{DOPT}(\mathcal{S}, s_0, G)$.

V obecném modelu mohou být jednotlivým operátorům $\varphi \in \Phi$ přiřazeny jejich *ceny* $c(\varphi)$, které vyjadřují náročnost jednotlivých operátorů (například časovou či finanční). Potom můžeme hovořit nejenom o délce řešení $(\varphi_1, \varphi_2, \dots, \varphi_n)$, ale také o jeho *celkové ceně* dané součtem cen všech operátorů daného řešení

$$\sum_{i=1}^n c(\varphi_i).$$

V tomto případě můžeme hovořit o *cenově optimálním* řešení, které má mezi všemi řešeními nejmenší celkovou cenu. Jsou-li ceny všech operátorů pozitivní, což se obvykle předpokládá, potom není z teoretického hlediska velkého rozdílu mezi hledáním optimálního a cenově optimálního řešení. Proto se v dalším budeme zabývat pouze hledáním

řešení nejmenší délky a ponecháváme čtenáři, aby si formou cvičení příslušná zobecnění provedl sám.

Hledání optimálního řešení nějaké úlohy (\mathcal{S}, s_0, G) můžeme snadno převést na problém hledání nejkratší cesty v orientovaném grafu. Ke každému stavovému prostoru $\mathcal{S} = (S, \Phi)$ lze totiž snadno sestavit *orientovaný graf* $G = (S, \vec{E})^3$, ve kterém existuje orientovaná hrana $(s \rightarrow s') \in \vec{E}$ právě tehdy, jestliže existuje operátor $\varphi \in \Phi$ převádějící stav s do stavu s' , t.j. $\varphi(s) = s'$. V takovém případě nalezení nejkratší (orientované) cesty z s_0 do některého cílového uzlu odpovídá nalezení optimálního řešení.

Poznámka. Na tomto místě je vhodné si uvědomit, že existují-li dva různé operátory převádějící stav s do stavu s' , je z hlediska optimálního řešení jedno, který z těchto dvou operátorů bude použit. Jestliže bychom hledali cenově optimální řešení, potom samozřejmě musíme vybrat v takovéto nejednoznačné situaci operátor s nejnižší cenou.

1.4 Metody hledání řešení

Pro hledání nejkratší cesty v orientovaných i neorientovaných grafech existuje celá řada efektivních algoritmů. Podívejme se nejprve na jeden z nejjednodušších.

Algoritmus prohledávání orientovaných grafů

I. Inicializace. $s_0 \rightarrow Z_1, \emptyset \rightarrow Z_2$.

II. Cyklus. 1. **IF** $Z_1 = \emptyset$ **THEN STOP**.

2. Vyber *nějakým způsobem* uzel $s \in Z_1$.

3. **IF** $s \in G$ **THEN STOP**, **ELSE** pro všechny následníky⁴ s^* uzlu s proved':

IF $s^* \notin Z_1 \cup Z_2$ **THEN** $s^* \rightarrow Z_1$.

4. $s \rightarrow Z_2$.

5. **GOTO** 1.

³Připomeňme si, že orientovaný graf je definován množinou uzlů a množinou jeho (orientovaných) hran, t.j. uspořádaných dvojic uzlů $(u \rightarrow v)$, kde uzel u je nazýván počátkem hrany a uzel v je konec příslušné hrany.

⁴Následníkem uzlu s rozumíme každý uzel \tilde{s} takový, že $(s \rightarrow \tilde{s}) \in \vec{E}$.

Pomocí tohoto algoritmu sice projdeme nějakou cestou z počátečního uzlu do cílového uzlu, nicméně si ji nezaznamenáváme. Abychom mohli nalezenou cestu zrekonstruovat, budeme si při zápisu uzlu s do zásobníků⁵ Z_1 , Z_2 zaznamenávat ještě údaj o tom, z kterého uzlu jsme se do uzlu s dostali. Algoritmus ukončí práci v okamžiku, kdy vybírá ze zásobníku Z_1 cílový uzel. Je-li u tohoto uzlu zapsán též uzel s' , ze kterého jsme do vybíraného cílového uzlu přešli, máme tak vlastně najednou poslední dva uzly nalezené cesty. Uzel s' však musí být uložen v zásobníku Z_2 . Je-li u tohoto uzlu také zapsáno, ze kterého uzlu jsme se do něj dostali, dostáváme tak snadno uzel výsledné cesty, který je třetí od konce. Tímto způsobem můžeme zrekonstruovat celou nalezenou cestu, až k počátečnímu uzlu s_0 . Je tedy vhodné do příslušných zásobníků neukládat pouze jednotlivé uzly, ale dvojice uzlů, tak jak je naznačeno v následujícím algoritmu. Vzhledem k tomu, že do počátečního uzlu nepřecházíme z žádného jiného, můžeme buď u tohoto uzlu nechat druhou pozici prázdnou, nebo tam zaznamenat libovolný jiný uzel.

Obecný algoritmus prohledávání orientovaných grafů

I. Inicializace. $(s_0, s_0) \rightarrow Z_1, \emptyset \rightarrow Z_2$.

II. Cyklus. 1. **IF** $Z_1 = \emptyset$ **THEN STOP**.

2. Vyber *nějakým způsobem* dvojici $(s, s') \in Z_1$.

3. **IF** $s \in G$ **THEN STOP**, **ELSE** pro všechny následníky⁶ s^* uzlu s proved:

IF $s^* \notin Z_1 \cup Z_2$ **THEN** $(s^*, s) \rightarrow Z_1$.

4. $(s, s') \rightarrow Z_2$.

5. **GOTO** 1.

Chování tohoto algoritmu zcela samozřejmě závisí na tom **jakým způsobem** vybíráme v kroku 2 dvojici uzlů (s, s') ze zásobníku Z_1 . Jestliže použijeme některou z klasických strategií jako např. **FIFO**⁷ nebo **LIFO**⁸, potom hovoříme o tak zvaném *slepém prohledávání*. Použití strategie FIFO vede k prohledávání grafu *do šířky*, naopak použití strategie LIFO vede k prohledávání *do hloubky*. Jestliže použijeme pro výběr dvojice ze zásobníku nějakou strategii závisující na okolnostech, které nejsou samotným algoritmem postižitelné, potom hovoříme o *řízeném prohledávání*.

⁵V tomto textu používáme pojem zásobníku obecně; v počítačové literatuře se někdy zásobníkem rozumí pouze struktura s výběrem LIFO - viz dále.

⁶Následníkem uzlu s rozumíme každý uzel \bar{s} takový, že $(s \rightarrow \bar{s}) \in \vec{E}$.

⁷First In First Out

⁸Last In First Out

1.5 Řízené prohledávání grafů

Pokusme se nyní sestavit funkci definovanou na uzlech uvažovaného grafu, pomocí jejíchž hodnot budeme řídit prohledávání grafu tak, aby algoritmus našel optimální řešení efektivním způsobem. Tím rozumíme že se budeme snažit vyhnout zdlouhavému prohledávání zbytečných cest, tak jak to dělá algoritmus prohledávání grafu do šířky.

Uvažujme na chvíli jednu úlohu (\mathcal{S}, s_0, G) . Pro ni definujme funkci

$$f : S \longrightarrow \{0, 1, 2, \dots\}$$

vyjadřující délku nejkratšího řešení procházejícího stavem s . Tuto funkci můžeme definovat jako součet délek optimálních řešení dvou podúloh

$$f(s) = \text{DOPT}(\mathcal{S}, s_0, \{s\}) + \text{DOPT}(\mathcal{S}, s, G).$$

Z této definice je vidět, že funkce f přiřazuje všem uzlům s ležícím na optimální cestě (či cestách) stejnou hodnotu

$$f(s) = f(s_0) = \text{DOPT}(\mathcal{S}, s_0, G).$$

Kdybychom tedy uměli vypočítat hodnotu funkce f pro všechny stavy dané úlohy, potom bychom mohli řídit výběr dvojice uzlů (s, s') ze zásobníku Z_1 tím, že bychom vždy vybírali tu dvojici (s, s') s nejnižší hodnotou funkce $f(s)$.

Z této myšlenky vychází níže uvedený heuristický algoritmus. Samozřejmě, že nemůžeme předpokládat znalost funkce f . Nicméně první část této funkce

$$\text{DOPT}(\mathcal{S}, s_0, \{s\})$$

můžeme odhadnout pomocí funkce $g(s)$, kterou počítáme v průběhu práce algoritmu. Odhad druhé části

$$\text{DOPT}(\mathcal{S}, s, G)$$

pak svěřujeme funkci $h(s)$, tak zvané *heuristické funkci*, o které předpokládáme, že její hodnoty počítáme na základě nějakých dodatečných znalostí týkajících se dané úlohy (heuristik). Proto výpočet této funkce není součástí algoritmu, i když právě na ní závisí celkové chování algoritmu. Obvykle se snažíme, aby funkce h byla dolním odhadem funkce $\text{DOPT}(\mathcal{S}, s, G)$, tedy aby

$$h(s) \leq \text{DOPT}(\mathcal{S}, s, G).$$

V tom případě totiž níže uvedený algoritmus \mathbf{A}^* nalezne vždy optimální řešení (viz Tvrzení 1). Ovšem v praktických situacích často slevíme z požadavku na optimalitu řešení výměnou za podstatné zvýšení efektivity výpočtu. Toho můžeme dosáhnout právě vhodnou volbou heuristické funkce.

Heuristický algoritmus A^*

I. Inicializace. $(s_0, s_0, 0) \rightarrow Z_1, \emptyset \rightarrow Z_2$.

II. Cyklus. 8

IF $Z_1 = \emptyset$ **THEN STOP.**

2. Vyber ze zásobníku Z_1 trojici $(s, s', g(s))$ s nejnižší hodnotou $g(s) + h(s)$.
3. **IF** $s \in G$ **THEN STOP, ELSE** pro všechny následníky s^* uzlu s proved' následující kroky (a) a (b).
 - (a) **IF** existuje v $Z_1 \cup Z_2$ trojice $(s^*, \tilde{s}, g(s^*))$ taková, že $g(s^*) > g(s) + 1$ **THEN** vyřad' $(s^*, \tilde{s}, g(s^*))$ z příslušného zásobníku.
 - (b) **IF** $(s^*, \tilde{s}, g(s^*)) \notin Z_1 \cup Z_2$ **THEN** $(s^*, s, g(s) + 1) \rightarrow Z_1$.
4. $(s, s', g(s)) \rightarrow Z_2$.
5. **GOTO** 1.

Tvrzení 1. Heuristický algoritmus A^* nalezne optimální řešení vždy, když použitá heuristická funkce $h(s)$ je dolním odhadem $\text{DOPT}(\mathcal{S}, s, G)$, neboli je-li pro všechna $s \in \mathcal{S}$

$$h(s) \leq \text{DOPT}(\mathcal{S}, s, G).$$

Důkaz. Označme si (s_0, s_1, \dots, s_n) optimální cestu z s_0 do nejbližšího cílového uzlu $s_n \in G$. Vzhledem k tomu, že předpokládáme, že $h(s)$ je dolním odhadem $\text{DOPT}(\mathcal{S}, s, G)$, je zřejmé, že pro všechny uzly (s_0, s_1, \dots, s_n) uvažované optimální cesty je

$$h(s_j) + j \leq \text{DOPT}(\mathcal{S}, s_0, G).$$

Algoritmus končí v kroku 3 poté, co byla ze zásobníku Z_1 vybrána nějaká trojice $(s, s', g(s))$ taková, že $s \in G$. Délka nalezeného řešení je rovna $g(s)$. Dokazujeme požadované tvrzení sporem. Předpokládejme tedy, že nalezené řešení není optimální, pak

$$g(s) > \text{DOPT}(\mathcal{S}, s_0, G) \geq h(s_j) + j$$

pro všechny uzly optimální cesty (s_0, s_1, \dots, s_n) . Vzhledem k tomu, že při prvním průběhu cyklu byly do zásobníku Z_1 uloženy všechny uzly, do kterých vede z s_0 hrana, musel tam být uložen i uzel s_1 ve tvaru trojice

$$(s_1, s_0, g(s_1) = 1).$$

Jelikož $g(s) > h(s_1) + 1$, algoritmus nemůže vybrat ze Z_1 trojici $(s, s', g(s))$ dříve, než uvedenou trojici $(s_1, s_0, g(s_1) = 1)$, která se do zásobníku dostala již v prvním kroku. To znamená, že dříve než může nastat námi uvažovaný konec algoritmu, musí se do zásobníku Z_1 dostat i všechny uzly, které jsou následníky uzlu s_1 . Tedy i uzel s_2 je ukládán do Z_1 před ukončením činnosti algoritmu ve tvaru trojice

$$(s_2, s_1, g(s_2) = 2).$$

Opět, vzhledem k dříve ukázané nerovnosti $g(s) > h(s_2) + 2$ musí být tato trojice dříve vybrána ze Z_1 než $(s, s', g(s))$ a tedy i trojice

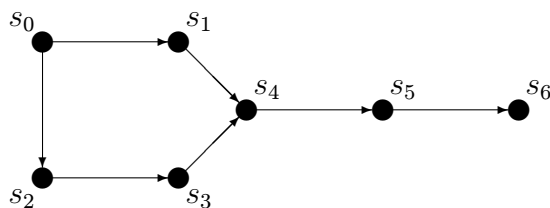
$$(s_3, s_2, g(s_3) = 3)$$

musí být uložena do zásobníku před ukončením činnosti algoritmu. Tímto způsobem můžeme pokračovat, dokud nedokážeme, že i trojice

$$(s_n, s_{n-1}, g(s_n) = n)$$

je vybírána z Z_1 dříve než $(s, s', g(s))$, což je spor, neboť algoritmus ukončí svou činnost při prvním výběru trojice ze zásobníku Z_1 obsahující uzel z cílové množiny. \square

Pozorný čtenář si jistě povšiml rozdílu mezi algoritmem \mathbf{A}^* a dříve uvedeným obecným algoritmem pro prohledávání grafů, který tkví ve skutečnosti, že algoritmus \mathbf{A}^* umožňuje (na rozdíl od obecného algoritmu), aby již jednou do zásobníku Z_2 uložený uzel byl ze zásobníku vyjmut a znovu “zpracován”. Nutnost tohoto opatření je zřejmá při aplikaci algoritmu \mathbf{A}^* na graf z následujícího obrázku.



při volbě počátečního uzlu s_0 , cílového uzlu s_6 a hodnotách heuristické funkce

$$h(s_0) = h(s_2) = h(s_3) = h(s_4) = h(s_5) = h(s_6) = 0, h(s_1) = 3.$$

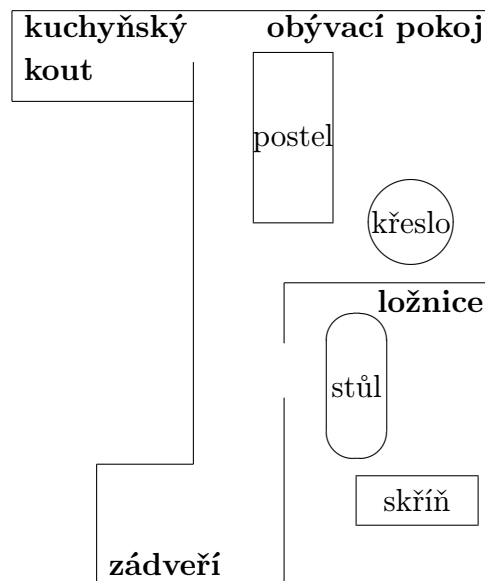
Postup práce algoritmu \mathbf{A}^* pro tuto situaci je znázorněn v tabulce na následující straně.

Poznámka. V kroku 4 je hodnota $g(s) + h(s)$ shodná pro oba uzly s obsažené v zásobníku Z_1 . Z teoretického hlediska je jedno, který z nich ze zásobníku vybereme. Zde jsme zvolili strategii LIFO.

krok	zásobník Z_1	$g(s) + h(s)$	zásobník Z_2	$g(s) + h(s)$
0	$(s_0, s_0, 0)$	0		
1	$(s_1, s_0, 1)$	4	$(s_0, s_0, 0)$	0
	$(s_2, s_0, 1)$	1		
2	$(s_1, s_0, 1)$	4	$(s_0, s_0, 0)$	0
	$(s_3, s_2, 2)$	2	$(s_2, s_0, 1)$	1
3	$(s_1, s_0, 1)$	4	$(s_0, s_0, 0)$	0
	$(s_4, s_3, 3)$	3	$(s_2, s_0, 1)$	1
			$(s_3, s_2, 2)$	2
4	$(s_1, s_0, 1)$	4	$(s_0, s_0, 0)$	0
	$(s_5, s_4, 4)$	4	$(s_2, s_0, 1)$	1
			$(s_3, s_2, 2)$	2
			$(s_4, s_3, 3)$	3
5	$(s_1, s_0, 1)$	4	$(s_0, s_0, 0)$	0
	$(s_6, s_5, 5)$	5	$(s_2, s_0, 1)$	1
			$(s_3, s_2, 2)$	2
			$(s_4, s_3, 3)$	3
			$(s_5, s_4, 4)$	4
6	$(s_6, s_5, 5)$	5	$(s_0, s_0, 0)$	0
	$(s_4, s_1, 2)$	2	$(s_2, s_0, 1)$	1
			$(s_3, s_2, 2)$	2
			$(s_5, s_4, 4)$	4
			$(s_1, s_0, 1)$	4
7	$(s_6, s_5, 5)$	5	$(s_0, s_0, 0)$	0
	$(s_5, s_4, 3)$	3	$(s_2, s_0, 1)$	1
			$(s_3, s_2, 2)$	2
			$(s_1, s_0, 1)$	4
			$(s_4, s_1, 2)$	2
8	$(s_6, s_5, 4)$	4	$(s_0, s_0, 0)$	0
			$(s_2, s_0, 1)$	1
			$(s_3, s_2, 2)$	2
			$(s_1, s_0, 1)$	4
			$(s_4, s_1, 2)$	2
			$(s_5, s_4, 3)$	3
9			$(s_0, s_0, 0)$	0
			$(s_2, s_0, 1)$	1
			$(s_3, s_2, 2)$	2
			$(s_1, s_0, 1)$	4
			$(s_4, s_1, 2)$	2
			$(s_5, s_4, 3)$	3
			$(s_6, s_5, 4)$	4

1.6 Cvičení

1. Navrhňte algoritmus, který po skončení práce obecného algoritmu prohledávání orientovaných grafů sestrojíte nalezenou orientovanou cestu. (*Návod: Využijte obsahu zásobníku Z_2 po skončení práce algoritmu; viz předchozí tabulka.*)
2. Dokažte, že slepé prohledávání do šířky vždy nalezne optimální cestu z počátku do některého cílového uzlu. (*Návod: Použijte Tvzení 1.*)
3. Sestrojte příklad ilustrující, že slepé prohledávání do hloubky nemusí nalézt optimální cestu.
4. Dokažte, že algoritmus \mathbf{A}^* realizuje prohledávání grafu do šířky, jestliže $h(s) = 0$ pro všechna $s \in S$.
5. Uvažujte známou hříčku o převozníkovi, jehož úkolem je převést vlka, kozu a hlávku zelí přes řeku. Na loďku se s ním vejde pouze jeden převážený objekt. Přitom nesmí ponechat bez dozoru na stejném břehu ani kozu se zelím, ani vlka s kozou. Navrhňte formalizaci této úlohy pomocí stavového prostoru a příslušných operátorů. Navrhňte heuristickou funkci a řešte úlohu pomocí algoritmu \mathbf{A}^* .
6. Po malování dvougarsionéry na sídlišti zůstal nábytek rozmístěn v místnostech tak, jak je vidět ze schematu.

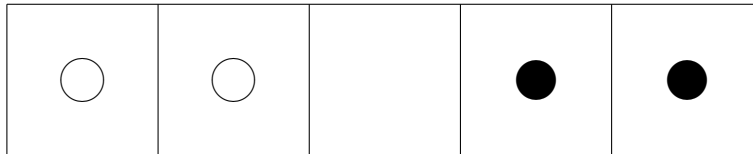


Úkolem je umístit stůl a křeslo do obývacího pokoje a do ložnice postel a skříň. Přičemž je třeba vzít do úvahy následující omezení vyplývající z velikosti bytu:

- Do kuchyňského koutu se z uvedeného nábytku vejde pouze skříň.
- Do obývacího pokoje i do ložnice se vejdou najednou pouze dva kusy nábytku.
- Křeslo neprojde dveřmi do ložnice.
- Do zádveří se vejde pouze křeslo.
- Při přemísťování nábytku musí být chodbička volná.

Navrhněte formalizaci této úlohy pomocí stavového prostoru a příslušných operátorů. Nalezněte řešení pomocí některého z algoritmů pro hledání cest v grafech.

7. Uvažujte hrací plochu o pěti polích obsazených dvěma bílými a dvěma černými hracími kameny dle následujícího obrázku.



Úkolem je přemístit černé a bílé hrací kameny navzájem. Do každého pole možno umístit pouze jeden hrací kámen. Jsou povoleny pouze dva druhy přesunu kamenů. Kámen se může buď pouze posunout na sousední volné pole, nebo smí přeskočit jeden kámen jiný kámen, je-li za ním volné pole.

Navrhněte formalizaci této úlohy pomocí stavového prostoru a příslušných operátorů. Nalezněte optimální řešení pomocí některého z algoritmů pro hledání cest v grafech a zdůvodněte jeho optimalitu.

1.7 Použitá literatura

1. Havel, I.M.: Robotika. *SNTL, Praha, 1972.*
2. Nilson, N.J.: Learning Machines. *Mc Graw Hill, New York, 1965.*
3. Nilson, N.J.: Methods of Artificial Intelligence. *Mc Graw Hill, New York, 1971.*

Kapitola 2

Obecný rozhodovací problém

Při formálním popisu rozhodovacího problému můžeme vycházet z několika různých modelů. Pro účely tohoto textu se budeme přidržovat nejjednoduššího z nich, ve kterém není uvažován čas. Takovouto typickou situaci si budeme často ilustrovat následujícím příkladem.

Společnost LICHVA a.s. se zabývá krátkodobými půjčkami. S její činností je samozřejmě spojeno riziko, že organizace či fyzická osoba, se kterou společnost LICHVA a.s. uzavřela dohodu, peníze v určené době nevrátí. Je tedy nutné před poskytnutím každé půjčky zvážit konkrétní rizika a rozhodnout, zda půjčku poskytnout či nikoliv.

Jedním ze způsobů, jak může námi uvažovaná společnost postupovat, je požádat každého žadatele o vyplnění dotazníku, ve kterém bude žádající fyzická či právnická osoba specifikovat jak sebe, tak i účel, pro který půjčku potřebuje. Část takového dotazníku může obsahovat otázky obsažené v tabulce na následující stránce. Úkolem pak bude na základě vyplněného dotazníku rozhodnout, zda půjčku poskytnout či nikoliv. Jinými slovy řečeno, budeme hledat vhodné zobrazení všech možných způsobů vyplnění daného dotazníku do dvouprvkové množiny $\{poskytnout, neposkytnout\}$.

Co rozumět pod pojmem *vhodné zobrazení* se bude v průběhu našeho dalšího výkladu měnit a bude záviset na studovaném problému. Ze začátku budeme mít na paměti především možnost snadného použití zkonstruovaného pravidla. Samozřejmě, že hlavním cílem však bude nalézt pravidlo, jehož použití povede k docílení nejvyšších zisků, tedy pravidla, které *správně odhadne* riziko spojené s poskytovanou půjčkou.

Abychom si v dalším rozuměli, musíme zavést jednoznačný formální aparát.

Otázkou rozumíme libovolnou konečnou množinu $Q = \{r_1, \dots, r_{n_Q}\}$. Její prvky r_i budeme nazývat *odpověďmi*. Přestože počet odpovědí na jednu otázku není předem ome-

Výše požadované půjčky Kč

Právnícká osoba ANO NE

Účel půjčky

Nákup kola ANO NE

Nákup auta ANO NE

Nákup výr. zařiz. ANO NE

Nákup nábytku ANO NE

.
.

.

Vyplní pouze právnické osoby

Právní forma s.r.o. a.s. družstvo

Počet zaměstnanců do 10 do 100 nad 100

Sídlo v Praze ANO NE

.
.

.

Vyplní pouze fyzické osoby

Pohlaví žena muž

Věk do 30 let do 50 let nad 50 let

Ženatý, vdaná ANO NE

.
.

.

zen, budeme vždy předpokládat, že není vysoký. Často budeme používat *binární* či *dichotomické* otázky, které mají pouze dvě možné odpovědi; například $\{ANO, NE\}$, $\{0, 1\}$, $\{+, -\}$, $\{mu, ena\}$ a podobně. Pojmem *detektor* budeme nazývat uvažovanou konečnou množinu otázek.

Podobně budeme pojmem *alternativa* označovat konečnou množinu *rozhodnutí* $\{a_1, \dots, a_m\}$. Řešení našeho úkolu bude reprezentovat *rozhodovací funkce* d , která přiřazuje každé kombinaci odpovědí nějaké rozhodnutí. Při detektoru $\{Q_1, \dots, Q_N\}$ a při alternativě $A = \{a_1, \dots, a_m\}$ je tedy rozhodovací funkcí každé zobrazení z kartézského součinu všech otázek do množiny všech rozhodnutí:

$$d: \times_{i=1}^N Q_i \longrightarrow A.$$

2.1 Rozhodovací tabulky

Jedním ze způsobů, jak reprezentovat znalost z dané oblasti tak, abychom ji mohli dále efektivně využít, je *rozhodovací tabulka*. Pro účely tohoto textu budeme vycházet z její nejjednodušší formy.

Rozhodovací tabulka obsahuje *sloupce* a *řádky*. Řádkům rozhodovací tabulky odpovídají otázky (horní část tabulky) a alternativy (spodní část tabulky). Sloupce slouží k záznamu částečných znalostí – pravidel. Každé políčko tabulky (t.j. průsečík sloupce a řádky) může být buď prázdné, nebo může obsahovat některou odpověď na příslušnou otázku (jedná-li se o řádku, které odpovídá otázka) nebo rozhodnutí (odpovídá-li řádce alternativa).

Výše požadované půjčky	5.000	2.000.000	5.000	100.000	2.000.000
Právnícká osoba	NE	ANO	ANO		NE
Nákup kola	ANO			ANO	
Nákup auta					ANO
Nákup výr. zaříz.		ANO	ANO		NE
Počet zaměstnanců			nad 100		
Věk					do 30
Půjčit	+	+		-	-

V uvedené rozhodovací tabulce například první sloupec obsahuje doporučení půjčit fyzické osobě, jestliže žádá o půjčku (zaokrouhleně) 5.000 Kč na nákup jízdního kola.

Naopak poslední sloupec nedoporučuje půjčku fyzické osobě věku do 30 let, jestliže žádá 2.000.000 Kč na nákup automobilu.

Poznámka. Z uvedeného příkladu vidíme, že prázdná mohou být i políčka v řádkách odpovídajících alternativám. To je především proto, že jedna rozhodovací tabulka může v dolní části obsahovat několik řádek, kterým odpovídá několik různých alternativ. Některé sloupce tedy mohou vypovídat pouze o některých alternativách.

Uvažujme konkrétní odpověď r na otázku Q . Jestliže určitý sloupec rozhodovací tabulky má v řádce odpovídajícím otázce Q buď zaznamenánu odpověď r , nebo je v této řádce prázdný, potom říkáme, že daný sloupec je *konzistentní* s odpovědí r . Nechť (r_1, \dots, r_N) jsou po řadě odpovědi na otázky (Q_1, \dots, Q_N) . Kombinace odpovědí (r_1, \dots, r_N) je konzistentní s nějakým sloupcem rozhodovací tabulky, jestliže každá z odpovědí r_i je s daným sloupcem konzistentní.

Jak jsme již řekli, rozhodovací funkce je funkce přiřazující každé kombinaci odpovědí (r_1, \dots, r_N) nějaké rozhodnutí $d(r_1, \dots, r_N) \in A$. Můžeme tedy říkat, že rozhodovací funkce d je *konzistentní* s danou rozhodovací tabulkou, jestliže pro každou kombinaci odpovědí (r_1, \dots, r_N) všechny sloupce, které jsou s ní konzistentní, obsahují v řádce odpovídající alternativě A buď rozhodnutí $d(r_1, \dots, r_N)$, nebo je tabulka v příslušném políčku prázdná.

Alternativa A je *rozlišitelná* v rozhodovací tabulce, jestliže existuje rozhodovací funkce

$$d : \times_{i=1}^N Q_i \longrightarrow A,$$

která je konzistentní s uvažovanou rozhodovací tabulkou. Rozhodovací tabulka je *jednoznačná*, jestliže jsou v ní rozlišitelné všechny alternativy, kterým odpovídají některé řádky dané rozhodovací tabulky.

2.2 Rozhodovací stromy

Nyní budeme hledat způsob, jak nalézt vhodnou rozhodovací funkci konzistentní s danou rozhodovací tabulkou. Současně nás bude zajímat i způsob, jakým budeme moci uvažovanou rozhodovací funkci aplikovat, t.j. jak nalézt hodnotu funkce $d(r_1, \dots, r_N)$ i pro N rovné několika stovkám, tedy v situaci, kdy je vyloučeno zaznamenat rozhodovací funkci formou seznamu přiřazujícího každé kombinaci odpovědí příslušné rozhodnutí a to i v tom případě, kdy je příslušná rozhodovací tabulka relativně malá.

Vhodným nástrojem pro tyto účely bude *rozhodovací strom*. Tím rozumíme uspořádanou čtveřici (G, q, ρ, α) , kde

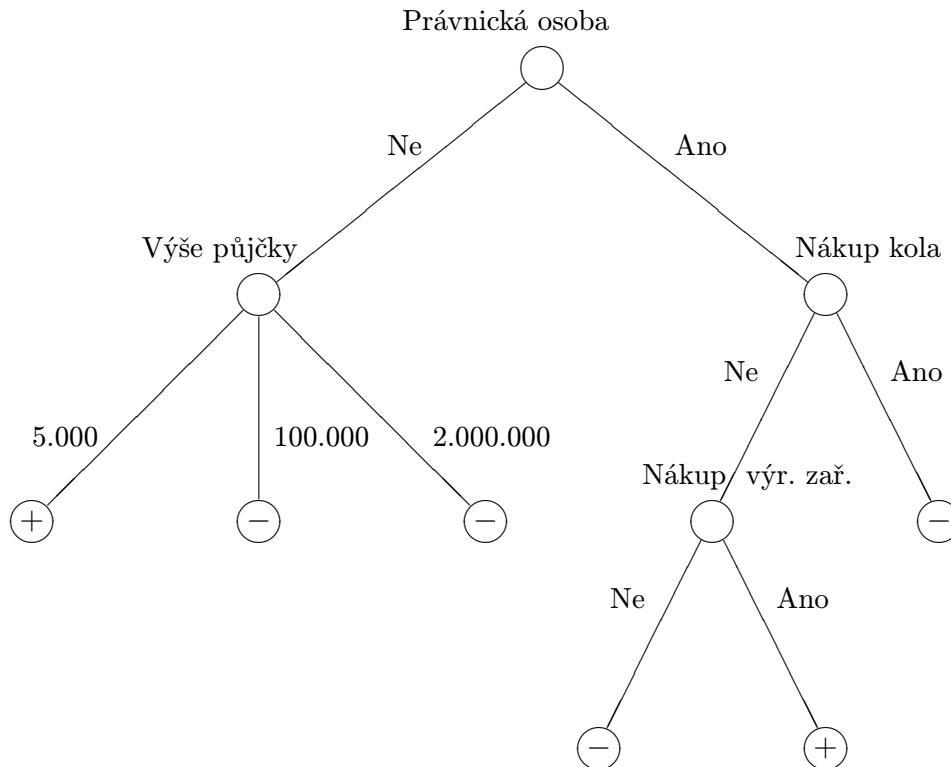
- G je kořenový graf typu strom¹,
- q je zobrazení přiřazující neterminálním uzlům stromu G otázky,
- ρ je zobrazení přiřazující hranám grafu G odpovědi na otázku,
- α je zobrazení přiřazující terminálním uzlům jednotlivá rozhodnutí.

Ovšem ne každá takováto čtveřice je rozhodovacím stromem. Tou je pouze čtveřice splňující následující 3 omezení.

1. Na žádné cestě vedoucí z kořenu do některého uzlu nejsou dva uzly, kterým by byla přiřazena stejná otázka.
2. Každý neterminální uzel u o úrovni $\ell(u)$ má tolik sousedů s úrovní $\ell(u) + 1$, kolik odpovědí má otázka přiřazená danému neterminálnímu uzlu.
3. Hranám spojujícím neterminální uzel u o úrovni $\ell(u)$ s jeho sousedy na úrovni $\ell(u) + 1$ jsou přiřazeny všechny odpovědi na otázku $q(u)$.

Příklad takového rozhodovacího stromu je uveden v následujícím obrázku. V něm jsou hodnoty příslušných zobrazení q, ρ napsány u jednotlivých uzlů či hran, hodnoty zobrazení α jsou zapsány přímo do kroužků označujících terminální uzly (listy) grafu.

¹Připomeňme si, že neorientovaný graf $G = (V, E)$ nazýváme *stromem*, jestliže je souvislý a neobsahuje cyklus. *Kořenovým stromem* je graf typu strom, jehož jeden uzel je nazván *kořenem*. Kořen stromu budeme obvykle značit u_0 . V kořenovém stromu má každý uzel v svoji *úroveň* $\ell(v)$, kterou je délka cesty z u_0 do v . Uzly stromu (s výjimkou kořenu), které mají pouze jednoho souseda, se nazývají *terminální*, nebo též *listy*. Ostatní uzly jsou *neterminální*. Kořen je listem pouze v případě, že celý strom je tvořen jediným uzlem u_0 .



Použití rozhodovacího stromu pro rozhodování vyplývá z jeho definice. Nejprve položíme otázku přiřazenou kořenu stromu. Vzhledem k tomu, že každá odpověď na tuto otázku je přiřazena některé hraně z uzlu vycházející (každá otázka právě jedné hraně), vybíráme zcela samozřejmě jako další otázku tu, která je přiřazena uzlu, který je s kořenem spojen příslušnou hranou. Opakováním tohoto postupu postupujeme vždy do uzlu o jednu úroveň vyšší, dokud nedosáhneme terminálního uzlu, kterému je přiřazeno rozhodnutí.

Pro každou kombinaci všech odpovědí $\{r_1, \dots, r_N\}$ můžeme tímto postupem nalézt

jednoznačně některé rozhodnutí (samozřejmě, že toto rozhodnutí nezáleží na těch odpovědích r_i , které otázky Q_i nebyly položeny). Vidíme tedy, že každému rozhodovacímu stromu odpovídá jednoznačně některá rozhodovací funkce. Budeme říkat, že rozhodovací strom tuto rozhodovací funkci *realizuje*.

Vraťme se nyní k rozhodovacím tabulkám. Pozorný čtenář si jistě všiml, že k jedné rozhodovací tabulce může existovat mnoho rozhodovacích funkcí, které jsou s ní konzistentní. Ke každé takové funkci může existovat (a také existuje) obrovské množství rozhodovacích stromů, které ji realizují. Máme-li znalosti obsaženy v rozhodovací tabulce, vzniká tedy přirozený problém, který z těch mnoha rozhodovacích stromů je nejlepší a jak jej zkonstruovat.

Každý z rozhodovacích stromů realizujících některou rozhodovací funkci konzistentní s danou rozhodovací tabulkou zahrnuje všechny znalosti v tabulce obsažené. Můžeme se tedy snažit vybrat takový rozhodovací strom, který na zaznamenání těchto znalostí vyžaduje nejméně místa, či strom, jehož použití při rozhodování je nejjednodušší. Zde se nabízejí dvě jednoduchá kritéria, podle kterých můžeme z uvažovaného souboru rozhodovacích stromů vybírat. Můžeme se snažit sestrojít strom, který má buď

(a) nejmenší počet uzlů, nebo

(b) nejkratší nejdelší cestu z kořenu do některého z listů.

Následující přímý algoritmus pro konstrukci rozhodovacího stromu z rozhodovací tabulky je navržen tak, aby zkonstruoval vždy strom realizující rozhodovací funkci konzistentní s výchozí tabulkou a přitom se snaží optimalizovat výsledný strom vzhledem k výše uvedenému kritériu (b). Nicméně, dosažení optimality v tomto směru tento algoritmus nezaručuje.

Přímý algoritmus pro konstrukci rozhodovacího stromu

I. Inicializace. Všem sloupcům rozhodovací tabulky přiřaď kořen u_0 ; $u_0 \rightarrow Z_1$.

II. Cyklus. 1. **IF** $Z_1 = \emptyset$ **THEN STOP.**

2. Vyjmi libovolný uzel $u \in Z_1$.

3. Uvažuj množinu R všech rozhodnutí obsažených ve sloupcích, kterým je přiřazen uzel u .

IF $R = \emptyset$ **THEN** u je terminální, přiřaď mu libovolné rozhodnutí. **GOTO** 1.

IF $|R| = 1$ **THEN** u je terminální, přiřaď mu rozhodnutí $r \in R$. **GOTO** 1.

IF $|R| > 1$ **THEN** u je neterminální.

4. Pro všechny otázky q , které nejsou přiřazeny žádnému uzlu cesty z u_0 do u proved':

Roztříd' sloupce, kterým je u přiřazen, na skupiny konzistentní s jednotlivými odpověďmi na otázku q .

Pro každou tuto skupinu sloupců spočti, kolik obsahují různých rozhodnutí.

Otázce přiřaď číslo $m(q)$ rovnající se maximálnímu počtu rozhodnutí v jedné skupině.

5. Vyber otázku q s nejmenší hodnotou $m(q)$.

Přiřaď q uzlu u .

Definuj tolik nových uzlů u_1, \dots, u_n kolik je odpovědí na otázku q . Dále definuj hrany $(u \rightarrow u_1), \dots, (u \rightarrow u_\alpha)$, kterým přiřaď pořadí všechny odpovědi na otázku q .

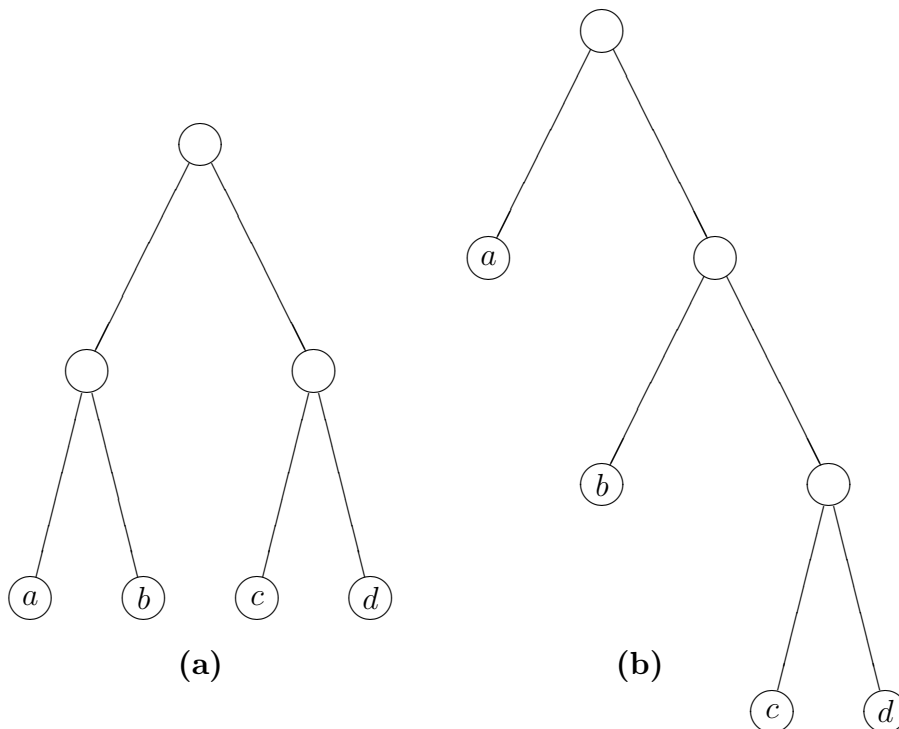
U všech sloupců, kterým byl přiřazen uzel u , přepiš u všemi těmi uzly, do nichž vedou z u hrany, jimž jsou přiřazeny odpovědi konzistentní s uvažovaným sloupcem.

6. $\{u_1, \dots, u_\alpha\} \rightarrow Z_1$.

7. **GOTO** 1.

2.3 Střední délka rozhodovacích stromů

Je zřejmé, že počet uzlů či délka nejdelší cesty rozhodovacího stromu sestrojeného výše uvedeným algoritmem závisí na výběru otázky přiřazované neterminálnímu uzlu v kroku 5. Pro zamezení nadměrného růstu délky větví slouží výpočet hodnoty $m(q)$ v kroku 4. Nicméně otázkou zůstává, zda délka nejdelší větve rozhodovacího stromu je skutečně tou hodnotou, která si zaslouží být minimalizována. Podívejme se na dva rozhodovací stromy na následujícím obrázku. Oba dva mají stejný počet uzlů a oba dva rozlišují rozhodnutí a, b, c, d . Rozhodovací strom označený jako **(b)** má delší cesty do listů, kterým jsou přiřazena rozhodnutí c a d , nicméně snadno si představíme situace, ve kterých použití právě tohoto rozhodovacího stromu bude výhodnější, než použití rozhodovacího stromu **(a)**. To nastane tehdy, jestliže při aplikaci rozhodovacího stromu převážné části rozhodovaných situací přísluší rozhodnutí a a situace, kterým přísluší rozhodnutí c či d jsou vzácné. Potom aplikace stromu **(b)** skončí ve většině případů po položení jediné otázky, zatímco použití stromu **(a)** vyžaduje vždy položení otázek dvou. Vzhledem k tomu, že nerovnoměrný výskyt jednotlivých rozhodnutí můžeme považovat za typický, budeme se nyní zabývat novým kritériem pro posuzování optimality rozhodovacích stromů, které budeme nazývat střední délkou rozhodovacího stromu.



Abychom měli zdroj, ze kterého získáme četnosti výskytu jednotlivých případů, budeme předpokládat, že nyní sestrojujeme rozhodovací strom na základě velkého datového souboru obsahujícího popis mnoha situací, které nastaly v minulosti. Na takovýto soubor se můžeme dívat jako na velkou rozhodovací tabulku, ve které jsou všechny řádky vyplněny ve všech sloupcích. V této kapitole budeme předpokládat, že se v souboru nevyskytují dva stejně popsané případy (t.j. stejné kombinace odpovědí na otázky), kterým by příslušela dvě různá rozhodnutí. Jinými slovy, předpokládáme, že námi uvažovaná velká rozhodovací tabulka rozlišuje uvažovanou alternativu.

S využitím takového velkého datového souboru můžeme získat odhady funkce, kterou budeme nazývat *pravděpodobností dosažení uzlů* rozhodovacího stromu. Uvažujme rozhodovací strom (G, q, ρ, α) . Každému uzlu grafu G odpovídá ta část datového souboru (t.j. sloupce myšlené rozhodovací tabulky), která je konzistentní s odpověďmi, které jsou přiřazeny hranám na cestě z kořenu u_0 do uvažovaného uzlu. Samotnému kořenu u_0 tedy odpovídá celý datový soubor. Kterémukoliv uzlu v , který je jeho přímý následník (tedy $\ell(v) = 1$), odpovídají ty případy, u kterých je na otázku $q(u_0)$ odpověď $\rho((u_0 \rightarrow v))$. Víme-li tedy, jaká část souboru odpovídá jednotlivým uzlům rozhodovacího stromu, můžeme vzít za *odhad* pravděpodobnosti dosažení uzlu $\pi(v)$ poměr počtu sloupců odpovídajících uzlu v k celkovému počtu sloupců v souboru.

Známe-li pravděpodobnosti dosažení uzlů $\pi(v)$ rozhodovacího stromu (G, q, ρ, α) můžeme spočítat jeho *střední délku*, která je definována výrazem

$$\bar{L}(G, q, \rho, \alpha) = \sum_{w \in W} \pi(w) \ell(w),$$

kde symbol W označuje množinu terminálních uzlů (listů) uvažovaného grafu G . Abychom se mohli začít zabývat metodami konstrukce rozhodovacích stromů optimálních ve smyslu právě zavedené střední délky, potřebujeme pro ni odvodit důležité nerovnosti.

Vzhledem k použití rozhodovacích stromů (či případně ke způsobu výpočtu odhadů funkce vyjadřující pravděpodobnost dosažení uzlů rozhodovacího stromu) je zřejmé, že pro pravděpodobnost dosažení kořenu platí

$$\pi(u_0) = 1.$$

Z libovolného neterminálního uzlu u rozhodovacího stromu postupujeme vždy do některého z jeho následníků a proto

$$\sum_{v \in \Gamma(u)} \pi(v) = \pi(u),$$

kde $\Gamma(u)$ označuje množinu přímých následníků uzlu u , t.j. uzlů, do nichž vede z u hrana a mají stupeň o jedna vyšší než u .

Z výše uvedených jednoduchých rovností dokážeme snadno matematickou indukcí (přes počet neterminálních uzlů grafu G) následující důležitou rovnost

$$\sum_{w \in W} \pi(w) = 1.$$

Je zajímavé, že analogické tři rovnosti, které jsme právě uvedli pro funkciv $\pi(v)$ splňuje i funkce $2^{-\ell(v)}$, pokud uvažujeme rozhodovací strom, jehož všechny otázky jsou binární.² Pro kořen u_0 stromu je vždy $\ell(u_0) = 0$ a proto

$$2^{-\ell(u_0)} = 1.$$

Uvažujme nyní nějaký neterminální uzel u rozhodovacího stromu (G, q, ρ, α) . Vzhledem k našemu předpokladu, že všechny otázky jsou binární, obsahuje množina $\Gamma(u)$ právě dva prvky a proto

$$\sum_{v \in \Gamma(u)} 2^{-\ell(v)} = 2 \cdot 2^{-(\ell(u)+1)} = 2^{-\ell(u)}.$$

Z těchto dvou rovností lze stejně jako pro pravděpodobnost dosažení uzlů π odvodit matematickou indukci i pro tuto funkci platnost rovnosti

$$\sum_{w \in W} 2^{-\ell(w)} = 1.$$

Rovnosti

$$\sum_{w \in W} \pi(w) = \sum_{w \in W} 2^{-\ell(w)} = 1$$

týkající se množiny terminálních uzlů využijeme pro odvození dolního odhadu střední délky rozhodovacího stromu. Pro toto odvození využijeme dvě nerovnosti používané často v teorii informace:

$$(a) \quad \mu_i \geq 0, \nu_i \geq 0, \sum_{i=1}^n \mu_i = \sum_{i=1}^n \nu_i = 1 \implies \sum_{i=1}^n \mu_i \log(\mu_i) \geq \sum_{i=1}^n \mu_i \log(\nu_i),$$

$$(b) \quad \mu_i \geq 0, \sum_{i=1}^n \mu_i = 1, \nu_i = \mu_i \text{ pro } i = 1, \dots, n-2 \text{ a } \nu_{n-1} = \mu_{n-1} + \mu_n \implies \\ \implies \sum_{i=1}^n \mu_i \log(\mu_i) \leq \sum_{i=1}^{n-1} \nu_i \log(\nu_i).$$

Dosadíme-li do první z nich za μ_i pravděpodobnosti dosažení listů $\pi(w)$ a za ν_i funkce $2^{-\ell(w)}$, dostáváme okamžitě nerovnost

$$\begin{aligned} H(\pi(w)) &= - \sum_{w \in W} \pi(w) \log(\pi(w)) \leq - \sum_{w \in W} \pi(w) \log(2^{-\ell(w)}) = \\ &= \log(2) \sum_{w \in W} \pi(w) \ell(w) = \log(2) \bar{L}(G, q, \rho, \alpha). \end{aligned}$$

²Mají-li všechny otázky stejný počet odpovědí různý od 2, potom stačí ve výrazu $2^{-\ell(v)}$ nahradit dvojku počtem odpovědí a odvozované vlastnosti zůstanou v platnosti.

Připomeňme si, že hodnota

$$H(\pi(w)) = - \sum_{w \in W} \pi(w) \log(\pi(w))$$

je Shannonovou entropií distribuce $\{\pi(w)\}_{w \in W}$.

Nechť alternativa rozlišovaná uvažovaným rozhodovacím stromem má jednotlivá rozhodnutí (a_1, \dots, a_m) . Jako jsme odhadovali pravděpodobnost dosažení uzlu rozhodovacího stromu na základě dat, můžeme odhadnout i pravděpodobnosti jednotlivých rozhodnutí $(p(a_1), \dots, p(a_m))$. Je zřejmé, že

$$\sum_{i=1}^m p(a_i) = 1.$$

a

$$p(a_i) = \sum_{w \in W: \alpha(w)=a_i} \pi(w),$$

z čehož několikanásobnou aplikací výše uvedené nerovnosti **(b)** dostáváme

$$H(p(a_i)) \leq H(\pi(w)) \leq \log(2) \bar{L}(G, q, \rho, \alpha).$$

Tímto jsme odvodili základní nerovnost sloužící jako dolní odhad střední délky rozhodovacího stromu, který má rozlišit rozhodnutí (a_1, \dots, a_m) . Touto nerovností je také zdůvodněn *princip maximálního snižování entropie*, který je často používán pro konstrukci rozhodovacích stromů. Pro jeho aplikaci stačí v přímém algoritmu pro konstrukci rozhodovacích stromů nahradit krok 4 následujícím postupem.

4. Pro všechny otázky q , které nejsou přiřazeny žádnému uzlu cesty z u_0 do u proved':
 Roztříd' sloupce, kterým je u připsán, na skupiny konzistentní s jednotlivými odpověďmi na otázku q .
- Pro každou tuto skupinu sloupců spočti, kolik obsahují různých rozhodnutí a spočti relativní četnosti těchto skupin.
- Otázce přiřaď číslo $m(q)$ rovnající se Shannonové entropii vypočtených relativních četností.

Takto získáme často používaný algoritmus, který je dosti efektivní, ale který nezaručuje sestavení optimálního rozhodovacího stromu (ve smyslu střední délky). Toho docílíme, použijeme-li nějakého silnějšího aparátu, například metody větví a mezí, kterou se začneme zabývat nyní.

2.4 Metoda větví a mezí

Jedná se o velice často používanou metodu pro hledání optimálního řešení nějaké úlohy a proto si ji zde popíšeme v abstraktní podobě. Později ji budeme velice stručně ilustrovat při použití pro konstrukci optimálního rozhodovacího stromu.

Předpokládejme, že naším úkolem je nalézt optimální řešení nějaké úlohy. Optimální zde znamená, že na množině τ všech řešení je definována nějaká kritériální funkce

$$C : \tau \longrightarrow \langle 0, +\infty \rangle,$$

jejíž hodnota $C(t)$ vyjadřuje “kvalitu” řešení $t \in \tau$. Za řešení optimální budeme považovat řešení, na kterém tato funkce nabývá svého minima. Pro účely tohoto textu předpokládáme, že množina všech řešení je konečná, nicméně natolik velká, že není možné zkonstruovat všechna řešení $t \in \tau$, vypočítat pro ně hodnotu kritériální funkce $C(t)$ a vybrat optimální řešení.

Popisovaný postup je založen na myšlence postupného rozkládání množiny všech řešení na menší skupiny, ze kterých pouze některé budou mít šanci obsahovat optimální řešení.

Rozkladem množiny τ rozumíme systém $\{\tau^1, \dots, \tau^k\}$ jeho disjunktních podmnožin, které jej pokrývají:

$$\begin{aligned} \tau^j &\subseteq \tau, \text{ pro všechna } j = 1, \dots, k, \\ \tau^i \cap \tau^j &= \emptyset, \text{ pro } i \neq j, \\ \bigcup_{j=1}^k \tau^j &= \tau. \end{aligned}$$

Nechť tedy $\mathcal{R}_1 = \{\tau_1^1, \dots, \tau_1^{k_1}\}$ je rozkladem množiny τ . Rozklad stejné množiny $\mathcal{R}_2 = \{\tau_2^1, \dots, \tau_2^{k_2}\}$ je zjemněním rozkladu \mathcal{R}_1 , jestliže pro každé $j = 1, \dots, k_1$ existují prvky $\tau_2^{l_1}, \dots, \tau_2^{l_m}$ ($l_m \geq 0$) rozkladu \mathcal{R}_2 takové, že

$$\tau_1^j = \bigcup_{l=l_1}^{l_m} \tau_2^l.$$

Předpokládejme, že pro každý prvek nějakého rozkladu $\mathcal{R} = \{\tau^1, \dots, \tau^k\}$ umíme spočítat dolní a horní odhady kritériální funkce

$$b(\tau^j) \leq C(t) \text{ pro všechna řešení } t \in \tau^j, \text{ a}$$

$$B(\tau^j) \geq C(t) \text{ pro všechna řešení } t \in \tau^j.$$

Potom je zřejmé, že existuje-li v takovém rozkladu \mathcal{R} prvek τ^j takový, že

- (i) $|\tau^j| = 1$ (t.j. τ^j je jednoprvková množina),
- (ii) $B(\tau^j) \leq b(\tau^i)$ pro všechna $i = 1, \dots, k$,

potom τ^j obsahuje optimální řešení.

Metoda větví a mezí je založena na konstrukci posloupnosti rozkladů $\mathcal{R}_1, \mathcal{R}_2, \dots$ takových, že \mathcal{R}_i je vždy zjemněním rozkladu \mathcal{R}_{i-1} . Vzhledem k tomu, že obvykle je tato posloupnost konstruována tak, že \mathcal{R}_i vznikne z rozkladu \mathcal{R}_{i-1} tím, že jednu množinu $\tau \in \mathcal{R}_{i-1}$ rozložíme na menší podmnožiny, odpovídá tento proces *větvení*. *Meze* z názvu metody jsou pak v každém kroku počítané dolní a horní odhady kriteriální funkce. Postup samozřejmě postupuje tak dlouho, dokud není zkonstruován rozklad, jehož jeden prvek splňuje vlastnosti (i) a (ii). Potom máme zaručeno, že v této množině existuje optimální řešení a postup končí.

Ilustrujme si nyní postup metody větví a mezí při konstrukci optimálního rozhodovacího stromu z datového souboru. Budeme i nadále předpokládat, že datový soubor tvoří rozhodovací tabulku, která je jednoznačná, neboli rozlišuje uvažovanou alternativu.

První konstruovaný rozklad $\mathcal{R}_1 = \{\tau_1^1, \dots, \tau_1^{k_1}\}$ bude v každé množině τ_1^i obsahovat rozhodovací stromy, jejichž kořenům je přiřazena stejná otázka

$$\tau_1^i = \{(G, q, \rho, \alpha) : q(u_0) = Q_i\}.$$

Jakým způsobem počítat dolní odhad střední délky stromů z množiny τ_1^i ? Uvažujme libovolný rozhodovací strom (G, q, ρ, α) z této množiny. Kdyby všechny uzly na úrovni 1 byly listy, potom by střední délka tohoto stromu byla 1. Jestliže však některá z odpovědí r na otázku $q(u_0)$ je konzistentní se sloupci tabulky, kterým jsou přiřazena rozdílná rozhodnutí, potom uzel u , pro který

$$\rho((u_0 \rightarrow u)) = r,$$

nemůže být listem ve stromu konzistentním s datovým souborem a musí na něj být “navěšen” rozhodovací podstrom konzistentní s tabulkou tvořenou jenom těmi sloupci, které jsou konzistentní s r . Pro omezení střední délky tohoto podstromu použijeme odhad

$$H(p_u(a)) \leq \log(2) \sum_{w \in W} \pi(w) \ell(w),^3$$

³ Jestliže na některou otázku uvažovaného detektoru je více otázek než dvě, potom 2 v tomto odhadu nahradíme maximálním počtem odpovědí na jednu otázku.

kde $p_u(a)$ je pravděpodobnostní (respektive četnostní) rozložení rozhodnutí ve sloupcích konzistentních s odpovědí $\rho((u_0 \rightarrow u)) = r$. Vzhledem k tomu, že tento podstrom přispívá k celkové střední délce uvažovaného rozhodovacího stromu pouze s váhou $p(u)$, mají všechny rozhodovací stromy z τ_1^i střední délku alespoň

$$b(\tau_1^i) = 1 + \sum_{u \in \Gamma(u_0)} \frac{p(u)}{\log(2)} H(p_u(a)).$$

Pro vytvoření rozkladu \mathcal{R}_2 vybereme z $\mathcal{R}_1 = \{\tau_1^1, \dots, \tau_1^{k_1}\}$ podmnožinu τ_1^i s nejnižší hodnotou $b(\tau_1^i)$. Nalezneme uzel $u \in \Gamma(u_0)$, pro který je $(p_u(a))$ pozitivní (v tom případě obsahují sloupce konzistentní s odpovědí $\rho((u_0 \rightarrow u))$ více než jedno rozhodnutí) a množinu τ_1^i rozdělíme do podmnožin tak, že každá podmnožina obsahuje rozhodovací stromy, ve kterých je uzlu u přiřazena stejná otázka. Rozklad $\mathcal{R}_2 = \{\tau_2^1, \dots, \tau_2^{k_2}\}$ je pak tvořen všemi množinami rozkladu $\mathcal{R}_1 = \{\tau_1^1, \dots, \tau_1^{k_1}\}$ s výjimkou množiny τ_1^i a množinami, na které jsme τ_1^i rozložili. Co mají všechny množiny rozkladu \mathcal{R}_2 společné je skutečnost, že začátky všech rozhodovacích stromů v každé podmnožině rozkladu jsou stejné. Můžeme tedy dolní odhady střední délky rozhodovacích stromů pro nové podmnožiny spočítat způsobem podobným, jako jsme počítali hodnoty $b(\tau_1^i)$.

Horní odhady střední délky rozhodovacích stromů obsažených v τ_j^i můžeme počítat pouze v případě, že množina obsahuje již pouze stromy konzistentní s daným datovým souborem; v tom případě společný začátek všech stromů je již sám stromem konzistentním s datovým souborem a my můžeme takovou množinu považovat za jednoprvkovou.

2.5 Cvičení

1. Ukažte, že alternativa A je rozlišitelná v dané rozhodovací tabulce, jestliže pro každé dva sloupce, které obsahují rozdílné rozhodnutí v řádce odpovídající A , existuje také řádek odpovídající nějaké otázce, který obsahuje v příslušných sloupcích různé odpovědi.
2. Navrhněte rozhodovací tabulku, s níž bude konzistentní více než jedna rozhodovací funkce. Umíte charakterizovat rozhodovací tabulky, ke kterým existuje jediná konzistentní rozhodovací funkce?
3. Uvažujte rozhodovací tabulku

Q_1	0	0	0	0	1	1
Q_2	0	1	1	1	1	0
Q_3	0	1	1	0	0	1
Q_4	0	0	1	1	0	0
Rozhodnutí	+	+	+	-	-	-

a sestrojte pro ni rozhodovací strom. Můžete říci něco o její optimalitě?

4. Ukažte, že jsou-li všechny otázky binární, má jakýkoliv rozhodovací strom vždy o jeden list více než neterminálních uzlů. (*Návod: Použijte matematickou indukci podle počtu uzlů grafu.*)
5. Jistě jste si uvědomili, že výše uvedený přímý algoritmus pro konstrukci rozhodovacích stromů sestrojí rozhodovací strom vždy, když vstupní rozhodovací tabulka rozlišuje uvažovanou alternativu. Navrhněte takovou úpravu tohoto algoritmu, aby mohl zpracovávat i rozhodovací tabulky, které tuto alternativu nerozlišují.
6. Ukažte, že algoritmus A^* je speciálním případem metody větví a mezí. (*Návod: Rozklady na množině řešení jsou vytvářeny tak, že každá podmnožina řešení je určena posloupností hran, které tvoří začátek všech řešení do podmnožiny patřící.*)

2.6 Literatura

1. Mitten, L.G.: Branch-and-bound methods: General formulations and properties. *Oper. Res.* 18(1970), 1, pp 24-34.
2. Payne, H.J., Meisel, W.S.: An algorithm for constructing optimal binary decision trees. *IEEE Trans. on Comp.*, 9.
3. Picard, C.F.: Graphs et Questionnaires. *Gauthier Villars, Paris, 1972.*
4. Chvalovský, V.: Rozhodovací tabulky. *SNTL, Praha, 1974.*
5. Jiroušek, R.: Heuristic method for construction of optimal sequential questionnaire. *Kybernetika*, 11(1975), pp 253-270.
6. Jiroušek, R.: An alternative method for construction of optimal sequential questionnaire. *Kybernetika*, 17(1981), pp 287-297.

Kapitola 3

Lineární rozhodovací funkce

Rozhodovací stromy jsou jedním typem efektivní reprezentace rozhodovacích funkcí. To je také jeden z hlavních důvodů, proč jsme se jim v minulé kapitole tolik věnovali. Nyní se budeme zabývat jiným způsobem, který může být v některých situacích ještě efektivnější. Pro jeho použití však budeme požadovat, aby odpovědi na všechny uvažované otázky byly číselné hodnoty, nebo aby byly pomocí čísel zakódovány. V tom případě si totiž můžeme jakoukoliv kombinaci všech odpovědí na N otázek (Q_1, \dots, Q_N) představit jako bod v N -dimenzionálním eukleidovském prostoru se všemi výhodami, které z toho plynou. Pak totiž můžeme uvažovat například eukleidovskou vzdálenost mezi dvěma kombinacemi odpovědí.

Uvažujme detektor (Q_1, \dots, Q_N) , jehož všechny odpovědi jsou reálná čísla a nějakou rozhodovací funkci

$$d : \times_{i=1}^N Q_i \longrightarrow \{a_1, a_2\},$$

rozlišující dvě rozhodnutí a_1 a a_2 . Říkáme, že tato rozhodovací funkce je *lineární*, jestliže existují reálná čísla $(w_1, \dots, w_N, w_{N+1})$ taková, že

$$d(r_1, \dots, r_N) = \begin{cases} a_1 & \text{jestliže } \sum_{i=1}^N r_i w_i < -w_{N+1}, \\ a_2 & \text{jestliže } \sum_{i=1}^N r_i w_i > -w_{N+1}. \end{cases}$$

Poznámka. Uvědomme si, že vzhledem k tomu, že stále uvažujeme konečný detektor s konečnými množinami odpovědí, je jedno, zda uvažujeme obě výše uvedené nerovnosti ostré, či zda v jedné z nich připustíme rovnost.

3.1 Rozhodnutí lineárně separabilní

V tomto odstavci si popíšeme metodu, která umožňuje nalézt požadované konstanty $(w_1, \dots, w_N, w_{N+1})$ v případě, že lineární rozhodovací funkce existuje, tedy v případě, kdy jsou rozhodnutí a_1 a a_2 lineárně separabilní. Podobně jako v minulé kapitole budeme předpokládat, že máme k dispozici datový soubor, který pro hledání lineární rozhodovací funkce použijeme.

Označme si M počet vektorů uvažovaného souboru. Každý vektor je tvořen N čísly, N odpověďmi na otázky (Q_1, \dots, Q_N) . Další symbolika se nám zjednoduší, jestliže si všechny tyto vektory prodloužíme o jednu souřadnici a jako $(N + 1)$. hodnotu každého vektoru dosadíme číslo 1. Uvažovaný datový soubor je tedy tvořen vektory Y_1, \dots, Y_M , z nichž každý má $N + 1$ souřadnic. O každém z nich předpokládáme, že víme, které rozhodnutí je pro něj správné, neboli předpokládáme znalost hodnot rozhodovací funkce $d(Y_i)$ pro všechna $i = 1, \dots, M$. Naším cílem je nalézt vektor W délky $N + 1$ takový, že

$$\begin{aligned} \langle W, Y_i \rangle < 0 &\iff d(Y_i) = a_1, \\ \langle W, Y_i \rangle > 0 &\iff d(Y_i) = a_2, \end{aligned}$$

kde symbolem $\langle W, Y_i \rangle$ značíme *skalární součin* vektorů W a Y ¹.

Hledaný vahový vektor $W = (w_1, \dots, w_{N+1})$ nalezneme pomocí následujícího jednoduchého algoritmu.

¹Skalární součin $\langle W, Y \rangle$ dvou vektorů $W = (w_1, \dots, w_{N+1})$ a $Y = (y_1, \dots, y_{N+1})$ je definován

$$\langle W, Y \rangle = \sum_{j=1}^{N+1} w_j y_j.$$

Výpočet lineární rozhodovací funkce

I. Inicializace. $W = (0, \dots, 0, 0)$; $i = nom = 0$.

II. Cyklus.

1. $i = i + 1$; $nom = nom + 1$; **IF** $i > M$ **THEN**
 $i = 1$.

2. **IF** $\langle W, Y_i \rangle \leq 0$ & $d(Y_i) = a_2$ **THEN**
 $W = W + Y_i$; $nom = 0$.

IF $\langle W, Y_i \rangle \geq 0$ & $d(Y_i) = a_1$ **THEN**
 $W = W - Y_i$; $nom = 0$.

3. **IF** $nom = M$ **THEN STOP ELSE GOTO 1.**

Tvrzení 2. Jestliže v datovém souboru jsou třídy přiřazené jednotlivým vektorům lineárně separabilní, t.j. existuje lineární rozhodovací funkce realizující rozhodnutí $d(Y_i)$ přiřazená vektorům daného datového souboru, potom výše uvedený algoritmus nalezne požadované řešení po konečném počtu kroků.

Důkaz. Definujme si nekonečnou posloupnost vektorů Y'_j pro $j = 1, 2, 3, \dots$ následujícím způsobem

$$Y'_j = \begin{cases} -Y_i & \text{pro } i = (j - 1 \bmod M) + 1 \text{ jestliže } d(Y_i) = a_1, \\ Y_i & \text{pro } i = (j - 1 \bmod M) + 1 \text{ jestliže } d(Y_i) = a_2. \end{cases}^2$$

Při takto definované posloupnosti algoritmus při modifikaci váhového vektoru W vektorem Y_i vlastně vždy přičítá vektor Y'_i a nikdy jej neodečítá. Přitom modifikace je prováděna v situacích, kdy $\langle W, Y'_i \rangle \leq 0$.

Předpokládejme, že existuje nějaký vektor W^* , který definuje odpovídající rozhodovací funkci d . Pro něj tedy

$$\langle W^*, Y_i \rangle < 0 \text{ jestliže } d(Y_i) = a_1$$

a

$$\langle W^*, Y_i \rangle > 0 \text{ jestliže } d(Y_i) = a_2.$$

Pro všechny vektory Y'_j tedy platí

$$\langle W^*, Y'_j \rangle > 0.$$

Důkaz požadovaného tvrzení provedeme sporem. Budeme tedy předpokládat, že tvrzení neplatí. V tom případě námi uvažovaný algoritmus nenalezne správný váhový vektor po konečném počtu kroků, což znamená, že z nekonečné posloupnosti Y'_1, Y'_2, \dots můžeme vybrat opět nekonečnou podposloupnost těch vektorů Y'_i , které v průběhu práce algoritmu modifikují počítaný váhový vektor. Označme si tuto vybranou podposloupnost $\bar{Y}_1, \bar{Y}_2, \dots$. Tu jsme tedy zkonstruovali tak, že platí (při použití $W_0 = (0, \dots, 0, 0)$):

$$\text{(i)} \quad W_i = W_{i-1} + \bar{Y}_i,$$

$$\text{(ii)} \quad \langle W_{i-1}, \bar{Y}_i \rangle \leq 0,$$

pro všechna přirozená čísla $i = 1, 2, \dots$. Proto z (i) dostáváme okamžitě

$$\text{(iii)} \quad W_i = \sum_{j=1}^i \bar{Y}_j.$$

Všechny tyto vlastnosti využijeme k odvození sporu.

² $(K \bmod M)$ označuje zbytek po celočíselném dělení čísla K číslem M .

Označme si

$$b = \min_{i=1,\dots,M} \langle Y'_i, W^* \rangle,$$

$$B = \max_{i=1,\dots,M} \| Y_i \|^2 .^3$$

Potom z linearity skalárního součinu dostáváme

$$\langle W_i, W^* \rangle = \sum_{j=1}^i \langle \bar{Y}_j, W^* \rangle \geq ib.$$

Z této nerovnosti, pomocí známé *Schwartzovy nerovnosti*

$$\langle W_i, W^* \rangle^2 \leq \| W_i \|^2 \| W^* \|^2,$$

dostáváme

$$\| W_i \|^2 \geq i^2 \frac{b^2}{\| W^* \|^2}.$$

Naproti tomu z rovnice

$$W_i = W_{i-1} + \bar{Y}_i$$

dostáváme

$$\| W_i \|^2 = \| W_{i-1} \|^2 + 2\langle W_{i-1}, \bar{Y}_i \rangle + \| \bar{Y}_i \|^2 \leq \| W_{i-1} \|^2 + \| \bar{Y}_i \|^2,$$

neboť víme, že

$$\langle W_{i-1}, \bar{Y}_i \rangle \leq 0.$$

Z toho můžeme odvodit nerovnosti

$$\begin{aligned} \| W_i \|^2 - \| W_{i-1} \|^2 &\leq \| \bar{Y}_i \|^2, \\ \| W_i \|^2 - \| W_{i-1} \|^2 + \| W_{i-1} \|^2 - \| W_{i-2} \|^2 &\leq \| \bar{Y}_i \|^2 + \| \bar{Y}_{i-1} \|^2, \\ \| W_i \|^2 - \| W_{i-1} \|^2 + \| W_{i-1} \|^2 - \| W_{i-2} \|^2 + \| W_{i-2} \|^2 - \| W_{i-3} \|^2 &\leq \\ &\leq \| \bar{Y}_i \|^2 + \| \bar{Y}_{i-1} \|^2 + \| \bar{Y}_{i-2} \|^2, \end{aligned}$$

a tak dále až dostaneme

$$\| W_i \|^2 - \| W_0 \|^2 = \| W_i \|^2 \leq \sum_{j=1}^i \| \bar{Y}_j \|^2 \leq iB^2,$$

neboť $\| W_0 \|^2 = 0$.

³ $\| Y \|^2$ označuje *normu* vektoru Y definovanou $\| Y \|^2 = \langle Y, Y \rangle$.

Odvodili jsme tedy nerovnosti

$$i^2 \frac{b^2}{\|W^*\|^2} \leq \|W_i\|^2 \leq iB^2,$$

které nemohou být splněny pro nekonečně mnoho čísel i , čímž jsme ukázali, že algoritmus musí skončit po konečném počtu kroků. \square

3.2 Příklady

Nyní si ukážeme dva příklady ilustrující práci uvedeného algoritmu.

1. příklad

Začněme s jednoduchým příkladem, kdy “datový” soubor (či rozhodovací tabulka) je dán čtyřmi vektory o dvou souřadnicích:

$$\begin{array}{c|cccc} Q_1 & 0 & 1 & 0 & 1 \\ Q_2 & 0 & 0 & 1 & 1 \\ \hline & - & - & + & + \end{array}$$

Průběh výpočtu algoritmu pro výpočet lineární rozhodovací funkce je v následující tabulce

krok	i	nom	W	Y_i	$\langle W, Y_i \rangle$	
0	0	0	(0,0,0)			
1	1	0	(0,0,0)	(0,0,1)	0	modifikace
2	2	1	(0,0,-1)	(1,0,1)	-1	
3	3	0	(0,0,-1)	(0,1,1)	-1	modifikace
4	4	1	(0,1,0)	(1,1,1)	1	
5	1	0	(0,1,0)	(0,0,1)	0	modifikace
6	2	1	(0,1,-1)	(1,0,1)	-1	
7	3	0	(0,1,-1)	(0,1,1)	0	modifikace
8	4	1	(0,2,0)	(1,1,1)	2	
9	1	0	(0,2,0)	(0,0,1)	0	modifikace
10	2	1	(0,2,-1)	(1,0,1)	-1	
11	3	2	(0,2,-1)	(0,1,1)	1	
12	4	3	(0,2,-1)	(1,1,1)	1	
13	1	4	(0,2,-1)	(0,0,1)	-1	

2. příklad

Zkusme, jak se zachová tento algoritmus, když s jeho pomocí začneme zpracovávat následující tabulku obsahující třídy lineárně neseparabilní.

$$\begin{array}{c|ccccc}
 Q_1 & 2 & 1 & 0 & 0 & 2 \\
 Q_2 & 1 & 0 & 1 & 2 & 0 \\
 Q_3 & 0 & 1 & 1 & 1 & 0 \\
 \hline
 & + & + & + & - & -
 \end{array}$$

Práce algoritmu pro výpočet lineární rozhodovací funkce je zaznamenán v následující tabulce

krok	i	nom	W	Y_i	$\langle W, Y_i \rangle$	
0	0	0	(0,0,0,0)			
1	1	0	(0,0,0,0)	(2,1,0,1)	0	modifikace
2	2	1	(2,1,0,1)	(1,0,1,1)	3	
3	3	2	(2,1,0,1)	(0,1,1,1)	2	
4	4	0	(2,1,0,1)	(0,2,1,1)	3	modifikace
5	5	0	(2,-1,-1,0)	(2,0,0,1)	4	modifikace
6	1	0	(0,-1,-1,-1)	(2,1,0,1)	-2	modifikace
7	2	1	(2,0,-1,0)	(1,0,1,1)	1	
8	3	0	(2,0,-1,0)	(0,1,1,1)	-1	modifikace
9	4	0	(2,1,0,1)	(0,2,1,1)	3	modifikace

Podíváme-li se na obsah této tabulky pozorněji, vidíme, že řádky obsahující 4. a 9. krok jsou (samozřejmě kromě čísla kroku) totožné. Znamená to tedy, že se výpočet zacyklil, což je v souladu s Tvzením 2. Zadané vektory jsou tedy skutečně lineárně neseparabilní.

3.3 Rozhodnutí lineárně neseparabilní

Dokázané tvrzení vypovídá o situaci, kdy existuje lineární rozhodovací funkce, která dovede správně zařadit všechny situace vyskytující se v datovém souboru. Ovšem otevřená je otázka, co dělat, když pro daný datový soubor taková lineární rozhodovací funkce neexistuje. V tom případě se musíme zamyslet, proč se snažíme nalézt lineární rozhodovací funkci pro uvažovaný problém, když neexistuje žádná ani pro daný datový soubor. V podstatě je třeba rozlišit dvě následující základní situace.

V případě, že datový soubor je dosti kvalitní a nepředpokládáme, že se v něm mohou vyskytovat chyby, je vhodné hledat překódování odpovědí na otázky tak, aby lineární rozhodovací funkce existovala. V tom případě doporučujeme uvažovat nejenom nelineární funkce jedné proměnné (jako například mocniny), ale i funkce několika proměnných. Znamená to, že doporučujeme překódovávat několik “souřadnic” najednou.

Často se však setkáme se situací, kdy musíme připustit, že neexistence lineární rozhodovací funkce je způsobena chybovostí v datech, tzv. *zašuměnými daty*. Není vzácností, že se v jednom datovém souboru vyskytnou dva případy popsané stejnou kombinací odpovědí na všechny otázky a přesto jsou jim připsána různá rozhodnutí. Potom můžeme hledat lineární (či nelineární) rozhodovací funkci, která se pro daný datový soubor dopustí nejmenšího možného počtu chyb. Obecně se tomuto problému začneme podrobně věnovat v následující kapitole. Zde si jenom uvedeme modifikaci výše uvedeného algoritmu, která je použitelná i na datové soubory, které nesplňují předpoklady výše uvedeného tvrzení.

Modifikovaný algoritmus pro výpočet lineární rozhodovací funkce

I. Inicializace. $W = (0, \dots, 0, 0)$; $i = n = nom = 0$.

II. Cyklus.

1. $i = i + 1$; $n = n + 1$; $nom = nom + 1$;

IF $i > M$ **THEN** $i = 1$.

2. **IF** $\langle W_{n-1}, Y_i \rangle \leq 0$ & $d(Y_i) = a_2$ **THEN**
 $W_n = W_{n-1} + \frac{1}{n} Y_i$; $nom = 0$.

IF $\langle W_{n-1}, Y_i \rangle > 0$ & $d(Y_i) = a_2$ **THEN**
 $W_n = W_{n-1}$.

IF $\langle W_{n-1}, Y_i \rangle \geq 0$ & $d(Y_i) = a_1$ **THEN**
 $W_n = W_{n-1} - \frac{1}{n} Y_i$; $nom = 0$.

IF $\langle W_{n-1}, Y_i \rangle < 0$ & $d(Y_i) = a_1$ **THEN**
 $W_n = W_{n-1}$.

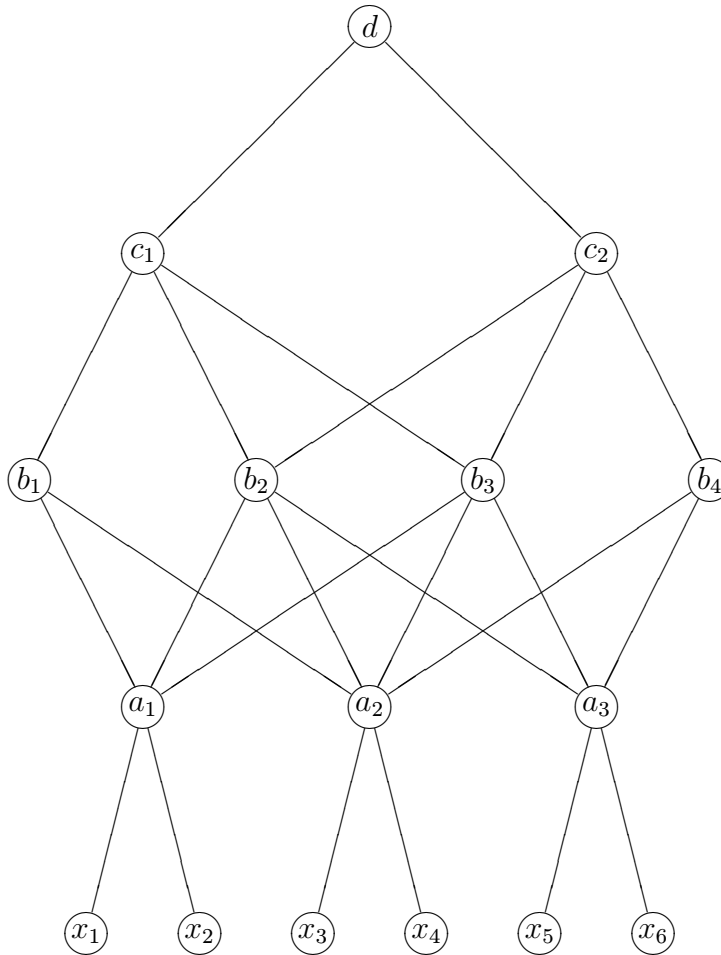
3. **IF** $nom = M$ **THEN STOP ELSE GOTO 1.**

U tohoto algoritmu si musíme uvědomit, že v případě, že lineární rozhodovací funkce

neexistuje, tak algoritmus neskončí příkazem **STOP** v kroku 3. Za výsledek výpočetního procesu považujeme váhový vektor W , ke kterému posloupnost počítaných váhových vektorů W_n konverguje. V praxi to znamená, že musíme proces nechat dostatečně dlouho počítat a jakmile budou změny již dostatečně malé, tak proces ukončíme. Musíme si však uvědomit, že uvažovaná modifikace výrazně prodlužuje časovou náročnost algoritmu.

3.4 Poznámka o neuronových sítích

Na tomto místě by mohla být zařazena rozsáhlá kapitola o neuronových sítích, které jsou vlastně složeny z lineárních neuronů - lineárních rozhodovacích funkcí. Není, neboť se v tomto textu z pochopitelných důvodů nemůžeme zabývat všemi metodami používanými pro zpracování znalostí v umělé inteligenci. Nicméně poznamenejme, že se jedná o poměrně silný a často používaný nástroj. Neuronová síť, zjednodušeně řečeno, je vlastně systém lineárních rozhodovacích funkcí vhodně propojených tak, že výstupy některých (t.j. jejich funkční hodnoty) tvoří vstupy (argumenty) jiných. Tak například na následujícím obrázku je nakreslena síť, ve které uzly a_1, a_2, a_3 jsou jednoduché lineární rozhodovací funkce každá o dvou argumentech. Výsledky všech těchto funkcí tvoří argumenty funkcí b_2 a b_3 . Funkce b_1, b_4 jsou, jak je patrné ze schématu, opět pouze dvouargumentové. Hodnoty vypočítané funkcemi b_1, b_2, b_3 , respektive b_2, b_3, b_4 jsou dále zpracovávány lineárními funkcemi c_1 respektive c_2 . Jejich výsledky pak tvoří vstup pro vlastní funkci d , která je neuronovou sítí z našeho obrázku realizována.



Není těžké ukázat, že takovýmto propojením lineárních funkcí dostáváme výsledně funkci, která již nemusí být lineární. Teorie neuronových sítí je v současné době již poměrně značně rozpracována. Jedním z nejdůležitějších výsledků, ukazující na obecnost tohoto přístupu, je právě teorem, který říká, že libovolnou funkci lze libovolně přesně aproximovat dostatečně bohatou neuronovou sítí.

Při studiu lineárních rozhodovacích funkcí jsme se zabývali možností automatického hledání příslušných koeficientů; tzv. učením lineární funkce. Poznamenejme, že i v oblasti neuronových sítí je pojem učení jeden z nejdůležitějších a že je mu věnována patřičná pozornost.

3.5 Cvičení

1. Nechť $\mathbf{r}^1 = (r_1^1, \dots, r_N^1)$ je soubor odpovědí na otázky (Q_1, \dots, Q_N) typický pro třídu a_1 . Obdobně, nechť $\mathbf{r}^2 = (r_1^2, \dots, r_N^2)$ je soubor odpovědí typický pro třídu a_2 . Uvažujte rozhodovací funkci $d(\mathbf{r})$

$$d : \times_{i=1}^N Q_i \longrightarrow \{a_1, a_2\},$$

takovou, že

$$d(\mathbf{r}) = \begin{cases} a_1 & \text{jestliže } \text{dist}(\mathbf{r}^1, \mathbf{r}) < \text{dist}(\mathbf{r}^2, \mathbf{r}), \\ a_2 & \text{jestliže } \text{dist}(\mathbf{r}^1, \mathbf{r}) > \text{dist}(\mathbf{r}^2, \mathbf{r}), \end{cases}$$

kde $\text{dist}(\mathbf{r}, \mathbf{r}')$ značí eukleidovskou vzdálenost bodů \mathbf{r} a \mathbf{r}' . Ukažte, že tato rozhodovací funkce je lineární.

2. Dokažte, že neexistuje lineární rozhodovací funkce konzistentní s následující rozhodovací tabulkou.

Q_1	0	2	2	1	2	0
Q_2	0	2	0	1	1	1
Q_3	0	2	0	1	1	2
	a_1	a_1	a_1	a_2	a_2	a_2

Použijte pro tuto situaci modifikovaný algoritmus uvedený v závěru kapitoly. Co můžete říci o vlastnostech lineární funkce, kterou získáte pomocí tohoto algoritmu?

3. Ukažte, že pro následující zrychlený algoritmus platí také Tvrzení 2.

Zrychlený algoritmus pro výpočet lineární rozhodovací funkce

I. Inicializace. $W = (0, \dots, 0, 0)$; $i = \text{nom} = 0$.

II. Cyklus.

1. $i = i + 1$; $\text{nom} = \text{nom} + 1$; **IF** $i > M$ **THEN**
 $i = 1$.

2. **IF** $d(Y_i) = a_2$ **THEN WHILE** $\langle W, Y_i \rangle \leq 0$
DO $W = W + Y_i$; $\text{nom} = 0$.

IF $d(Y_i) = a_1$ **THEN WHILE** $\langle W, Y_i \rangle \geq 0$
DO $W = W - Y_i$; $\text{nom} = 0$.

3. **IF** $\text{nom} = M$ **THEN STOP ELSE GOTO** 1.

4. Obdobně jako ve cvičení číslo 1 uvažujte reprezentanty $\mathbf{r}^i = (r_1^i, \dots, r_N^i)$ L tříd (a_1, \dots, a_L) . Uvažujte rozhodovací funkci $d(\mathbf{r})$ která opět přiřazuje vektoru $\mathbf{r} = (r_1, \dots, r_N)$ tu třídu a_i , jejíž reprezentant $\mathbf{r}^i = (r_1^i, \dots, r_N^i)$ je ve smyslu eukleidovské metriky nejbližší k \mathbf{r} . Nalezněte rozhodovací funkci, která tento způsob rozhodování realizuje.

Pokuste se zobecnit dosažený výsledek a definujte, kdy je rozhodovací funkce

$$d : \times_{i=1}^N Q_i \longrightarrow \{a_1, \dots, a_L\},$$

lineární.

3.6 Literatura

1. Nilson, N.J.: Learning Machines. *Mc Graw Hill, New York, 1965.*
2. Minsky, M., Popert, S.: Perceptrons. MIT Press, Cambridge, England, 1969.

Kapitola 4

Chyba rozhodování

V závěru minulé kapitoly jsme uvedli algoritmus pro hledání lineární rozhodovací funkce i v případě, že pro datový soubor, který je pro hledání rozhodovací funkce použit, taková rozhodovací funkce neexistuje. Znamená to, že jsme se smířili s myšlenkou, že navržená rozhodovací funkce nebude rozhodovat vždy správně. Ve skutečnosti situace, kdy můžeme nalézt funkci rozhodující bezchybně, jsou vzácné. Proto se nyní začneme zabývat vlastně nejdůležitější vlastností rozhodovacích funkcí - jejich chybou rozhodování.

4.1 Přípustné rozhodovací funkce

Abychom mohli chybu rozhodování definovat, budeme předpokládat, že známe pravděpodobnostní distribuci P definovanou na $(N + 1)$ -rozměrném kartézském součinu

$$\times_{i=1}^N Q_i \times \{a_1, a_2\}.$$

Obdobně jako v minulé kapitole začínáme tedy i zde s jednoduchou situací, kdy uvažované rozhodovací funkce rozlišují pouze dvě rozhodnutí.

Pro zjednodušení symboliky si označíme

$$\mathbf{Q} = \times_{i=1}^N Q_i.$$

Uvažovaná pravděpodobnostní distribuce P je tedy definována na

$$\mathbf{Q} \times \{a_1, a_2\}.$$

Pro každou rozhodovací funkci

$$d : \mathbf{Q} \longrightarrow \{a_1, a_2\},$$

rozlišující dvě rozhodnutí a_1 a a_2 můžeme definovat pravděpodobnosti chyb dvou druhů. První zahrnuje situace, kdy skutečnost odpovídá a_1 a rozhodovací funkce d navrhuje přijmout rozhodnutí a_2 . Tuto chybu označíme symbolem $p_d(a_1 \rightarrow a_2)$ a budeme ji počítat podle vzorce

$$\begin{aligned} p_d(a_1 \rightarrow a_2) &= \sum_{\mathbf{r} \in \mathbf{Q}: d(\mathbf{r})=a_2} P(\mathbf{r}|a_1) = \sum_{\mathbf{r} \in \mathbf{Q}} P(\mathbf{r}|a_1) \delta(d(\mathbf{r}), a_2) = \\ &= \sum_{\mathbf{r} \in \mathbf{Q}} P(\mathbf{r}|a_1) (1 - \delta(d(\mathbf{r}), a_1)), \end{aligned}$$

kde symbolem $\delta(a, b)$ ¹ budeme v této kapitole označovat funkci definovanou

$$\delta(a, b) = \begin{cases} 1 & \text{jestliže } a = b, \\ 0 & \text{jestliže } a \neq b. \end{cases}$$

Obdobně

$$p_d(a_2 \rightarrow a_1) = \sum_{\mathbf{r} \in \mathbf{Q}: d(\mathbf{r})=a_1} P(\mathbf{r}|a_2) = \sum_{\mathbf{r} \in \mathbf{Q}} P(\mathbf{r}|a_2) (1 - \delta(d(\mathbf{r}), a_2))$$

označuje pravděpodobnost chyby nastávající, když se dopouštíme chyby při použití rozhodovací funkce d a rozhodujeme se pro a_1 .

Zdůrazněme na tomto místě, že zavedené chyby $p_d(a_1 \rightarrow a_2)$ a $p_d(a_2 \rightarrow a_1)$ závisí nejen na (v indexu vyznačené) rozhodovací funkci d , ale též na pravděpodobnostní distribuci P , kterou nyní považujeme za pevně danou. Kdyby se tato distribuce změnila, změnily by se samozřejmě i obě tyto chyby.

Označme si \mathcal{D} rodinu všech uvažovaných rozhodovacích funkcí

$$\mathcal{D} = \{d : \mathbf{Q} \longrightarrow \{a_1, a_2\}\}.$$

Budeme říkat, že rozhodovací funkce $d_1 \in \mathcal{D}$ *dominuje* rozhodovací funkci $d_2 \in \mathcal{D}$, jestliže buď

$$p_{d_1}(a_2 \rightarrow a_1) \leq p_{d_2}(a_2 \rightarrow a_1) \text{ a současně } p_{d_1}(a_1 \rightarrow a_2) < p_{d_2}(a_1 \rightarrow a_2),$$

nebo

$$p_{d_1}(a_2 \rightarrow a_1) < p_{d_2}(a_2 \rightarrow a_1) \text{ a současně } p_{d_1}(a_1 \rightarrow a_2) \leq p_{d_2}(a_1 \rightarrow a_2).$$

¹tzv. Croneckerovo δ

Budeme říkat, že rozhodovací funkce $d \in \mathcal{D}$ je *přípustná* jestliže v \mathcal{D} neexistuje rozhodovací funkce, která by dominovala funkci d .

Příklad. Uvažujme $\mathbf{Q} = \{0, 1\}$. Potom existují čtyři rozhodovací funkce $\mathcal{D} = \{d_1, d_2, d_3, d_4\}$,

$$d_i : \{0, 1\} \longrightarrow \{a_1, a_2\}.$$

Jedná se o funkce

$$d_1(0) = a_1, \quad d_1(1) = a_1,$$

$$d_2(0) = a_1, \quad d_2(1) = a_2,$$

$$d_3(0) = a_2, \quad d_3(1) = a_1,$$

$$d_4(0) = a_2, \quad d_4(1) = a_2.$$

Z definice $p_{d_i}(a_j \rightarrow a_{3-j})$ dostáváme

	$p_{d_i}(a_1 \rightarrow a_2)$	$p_{d_i}(a_2 \rightarrow a_1)$
d_1	0	$P(0 a_2) + P(1 a_2)$
d_2	$P(1 a_1)$	$P(0 a_2)$
d_3	$P(0 a_1)$	$P(1 a_2)$
d_4	$P(0 a_1) + P(1 a_1)$	0

Je tedy zřejmé, že obě chyby skutečně závisí na uvažované distribuci P . Například pro distribuci P

P	a_1	a_2
0	0.2	0.3
1	0.1	0.4

jsou tyto chyby obsaženy v následující tabulce. Z ní vidíme, že rozhodovací funkce d_1, d_2 a d_4 jsou přípustné, zatímco funkce d_3 není přípustná, neboť je dominována funkcí d_2 .

pro P	$p_{d_i}(a_1 \rightarrow a_2)$	$p_{d_i}(a_2 \rightarrow a_1)$
d_1	0	1
d_2	$\frac{1}{3}$	$\frac{3}{7}$
d_3	$\frac{2}{3}$	$\frac{4}{7}$
d_4	1	0

Jestliže však místo distribuce P budeme uvažovat distribuci \tilde{P} ,

\tilde{P}	a_1	a_2
0	0.1	0.3
1	0.2	0.4

která se od P liší pouze prohozením hodnot v prvním sloupci, dostáváme

pro \tilde{P}	$p_{d_i}(a_1 \rightarrow a_2)$	$p_{d_i}(a_2 \rightarrow a_1)$
d_1	0	1
d_2	$\frac{2}{3}$	$\frac{3}{7}$
d_3	$\frac{1}{3}$	$\frac{4}{7}$
d_4	1	0

a tedy všechny čtyři rozhodovací funkce jsou přípustné.

Tvrzení 3. Jestliže existují pozitivní čísla $w(a_1)$ a $w(a_2)$ taková, že rozhodovací funkce

$$d(\mathbf{r}) = \begin{cases} a_1 & \text{jestliže } P(\mathbf{r}|a_1) > \frac{w(a_2)}{w(a_1)}P(\mathbf{r}|a_2), \\ a_2 & \text{jestliže } P(\mathbf{r}|a_1) < \frac{w(a_2)}{w(a_1)}P(\mathbf{r}|a_2), \end{cases}$$

potom je tato rozhodovací funkce přípustná.

Poznámka. Pozorný čtenář si jistě povšiml, že v uvedeném tvrzení se vyskytuje pouze poměr váhových vektorů ($w(a_2)/w(a_1)$) a že bychom tedy mohli vyžadovat existenci pouze jediného čísla. Nicméně, uvažování dvojice vah $w(a_1)$ a $w(a_2)$ vyplynulo z historie vývoje použitého statistického rozhodovacího modelu a navíc budeme přiřazovat vahám $w(a_1)$, $w(a_2)$ určitý význam.

Důkaz. Uvažujme sumu

$$\begin{aligned} & w(a_1)p_d(a_1 \rightarrow a_2) + w(a_2)p_d(a_2 \rightarrow a_1) = \\ & = \sum_{\mathbf{r} \in \mathbf{Q}} w(a_1)P(\mathbf{r}|a_1)\delta(d(\mathbf{r}), a_2) + \sum_{\mathbf{r} \in \mathbf{Q}} w(a_2)P(\mathbf{r}|a_2)\delta(d(\mathbf{r}), a_1) = \\ & = \sum_{\mathbf{r} \in \mathbf{Q}} [w(a_1)P(\mathbf{r}|a_1)\delta(d(\mathbf{r}), a_2) + w(a_2)P(\mathbf{r}|a_2)\delta(d(\mathbf{r}), a_1)], \end{aligned}$$

kde hodnoty $w(a_1)$ a $w(a_2)$ odpovídají předpokladům dokazovaného tvrzení. Vzhledem k tomu, že pro všechna $\mathbf{r} \in \mathbf{Q}$

$$\delta(d(\mathbf{r}), a_2) + \delta(d(\mathbf{r}), a_1) = 1,$$

přispívá do uvedené sumy každé $\mathbf{r} \in \mathbf{Q}$ buď hodnotou $w(a_1)P(\mathbf{r}|a_1)$ (je-li $d(\mathbf{r}) = a_2$) nebo hodnotou $w(a_2)P(\mathbf{r}|a_2)$ (v opačném případě). Z předpokladu

$$d(\mathbf{r}) = \begin{cases} a_1 & \text{jestliže } w(a_1)P(\mathbf{r}|a_1) > w(a_2)P(\mathbf{r}|a_2), \\ a_2 & \text{jestliže } w(a_1)P(\mathbf{r}|a_1) < w(a_2)P(\mathbf{r}|a_2), \end{cases}$$

tak vyplývá, že uvažovaná rozhodovací funkce minimalizuje hodnotu

$$w(a_1)p_d(a_1 \rightarrow a_2) + w(a_2)p_d(a_2 \rightarrow a_1).$$

Neexistuje tedy rozhodovací funkce d' , pro kterou by tato hodnota byla menší.

Jestliže si vzpomeneme na analytickou geometrii probíranou na střední škole, uvědomíme si, že hodnota

$$w(a_1)p_d(a_1 \rightarrow a_2) + w(a_2)p_d(a_2 \rightarrow a_1)$$

je úměrná vzdálenosti bodu

$$[p_d(a_1 \rightarrow a_2), p_d(a_2 \rightarrow a_1)]$$

od přímky procházející počátkem a mající směrnici $(-w(a_1)/w(a_2))$. Víme tedy, že nemůže existovat rozhodovací funkce d' , a jí odpovídající bod

$$[p_{d'}(a_1 \rightarrow a_2), p_{d'}(a_2 \rightarrow a_1)],$$

který by byl blíže k této přímce. Vzhledem k tomu, že přímka má zápornou směrnici, nemůže být pro žádnou rozhodovací funkci

$$p_{d'}(a_2 \rightarrow a_1) \leq p_d(a_2 \rightarrow a_1) \ \& \ p_{d'}(a_1 \rightarrow a_2) < p_d(a_1 \rightarrow a_2)$$

ani

$$p_{d'}(a_2 \rightarrow a_1) < p_d(a_2 \rightarrow a_1) \ \& \ p_{d'}(a_1 \rightarrow a_2) \leq p_d(a_1 \rightarrow a_2).$$

□

4.2 Optimalita rozhodovacích funkcí

Pomocí hodnot $w(a_1)$ a $w(a_2)$ můžeme také charakterizovat dvě speciální rozhodovací funkce mající mezi všemi přípustnými funkcemi vyjíměčné postavení. Jednou z nich

je *bayesovsky optimální* rozhodovací funkce, která minimalizuje *celkovou střední chybu rozhodování* definovanou

$$\begin{aligned} e_P(d) &= P(a_1)p_d(a_1 \rightarrow a_2) + P(a_2)p_d(a_2 \rightarrow a_1) = \\ &= \sum_{i=1}^2 \sum_{\mathbf{r} \in \mathbf{Q}} P(a_i)P(\mathbf{r}|a_i)(1 - \delta(d(\mathbf{r}), a_i)) = \\ &= \sum_{i=1}^2 \sum_{\mathbf{r} \in \mathbf{Q}} P(\mathbf{r}, a_i)(1 - \delta(d(\mathbf{r}), a_i)). \end{aligned}$$

Hledáme-li tuto rozhodovací funkci, pak volíme $w(a_1) = P(a_1)$ a $w(a_2) = P(a_2)$.

Rozhodovací funkci optimální vzhledem ke kritériu *maximální věrohodnosti* dostáváme, jestliže zvolíme $w(a_1) = w(a_2) = 1$. Potom je samozřejmě minimalizována hodnota

$$p_d(a_1 \rightarrow a_2) + p_d(a_2 \rightarrow a_1).$$

Jiný způsob výběru optimální rozhodovací funkce mezi přípustnými funkcemi budeme používat, jestliže budeme znát *rizika*, která podstupujeme při špatném rozhodování. V mnoha případech totiž nejsou chyby symetrické. Je zřejmé, že chyba které se dopustíme, jestliže půjčíme nespolehlivé osobě, která nám půjčku nevrátí, je zcela jiného charakteru než chyba odpovídající situaci, kdy nepůjčíme spolehlivé osobě. V takových situacích zavádíme hodnoty takzvané *ztrátové funkce*

$$\ell : \{a_1, a_2\} \times \{a_1, a_2\} \longrightarrow \langle 0, +\infty \rangle,$$

kteřou hodnotíme ztrátu nastávající v situaci, kdy správné rozhodnutí je a_1 a my se rozhodneme pro a_2 , či naopak. Potom můžeme definovat *celkové riziko* či *celkovou střední ztrátu* výrazem

$$\begin{aligned} R_P(d) &= \sum_{i=1}^2 \sum_{\mathbf{r} \in \mathbf{Q}} P(\mathbf{r}, a_i) \ell(a_i, d(\mathbf{r}))^2 \\ &= \sum_{i=1}^2 \sum_{\mathbf{r} \in \mathbf{Q}} P(a_i) \ell(a_i, d(\mathbf{r})) P(\mathbf{r}|a_i). \end{aligned}$$

Z výše uvedených výrazů okamžitě plyne, jak volit příslušné váhové vektory při konstrukci optimálních rozhodovacích funkcí. Za předpokladu, že $\ell(a, a) = 0$ je tedy optimální rozhodovací funkce určena opět ve známém tvaru

$$d_{opt}(\mathbf{r}) = \begin{cases} a_1 & \text{jestliže } \ell(a_1, a_2)P(\mathbf{r}|a_1) > \ell(a_2, a_1)P(\mathbf{r}|a_2), \\ a_2 & \text{jestliže } \ell(a_1, a_2)P(\mathbf{r}|a_1) < \ell(a_2, a_1)P(\mathbf{r}|a_2). \end{cases}$$

¹Samozřejmě, že ve většině případů volíme $\ell(a, a) = 0$.

4.3 Rozhodování za neurčitosti

Všechny výše uvedené úvahy byly ovšem založeny na předpokladu, že známe pravděpodobnostní rozložení popisující situaci, za které máme rozhodovat. Tento předpoklad je bohužel ve většině situací nereálný. Velice často se musíme spokojit s nepřesnou znalostí, která neurčuje příslušnou distribuci jednoznačně, ale vymezuje pouze, že “správná” pravděpodobnostní distribuce patří do nějaké třídy distribucí Π . Jak se zachováme v tomto případě?

Častým, nikoliv však nejlepším řešením bývá použití *principu maximální entropie* zvaného též *principem maximální neurčitosti*. Podle něj vybereme z Π distribuci P_H s nejvyšší Shannonovskou entropií

$$H(P_H) = \max_{P \in \Pi} H(P) = \max_{P \in \Pi} \left(- \sum_{i=1}^2 \sum_{\mathbf{r} \in \mathbf{Q}} P(\mathbf{r}, a_i) \log P(\mathbf{r}, a_i) \right)$$

a dále postupujeme, jako by distribuce P_H byla ta pravá.

Při hledání optimální rozhodovací funkce je však mnohem správnější použití tak zvaného *principu minimaxu*, který si nyní vysvětlíme.

Předpokládejme, že naším cílem je nalézt bayesovsky optimální rozhodovací funkci. Kdybychom znali distribuci P , snažili bychom se vybrat rozhodovací funkci d minimalizující hodnotu

$$e_P(d) = \sum_{i=1}^2 \sum_{\mathbf{r} \in \mathbf{Q}} P(\mathbf{r}, a_i) (1 - \delta(d(\mathbf{r}), a_i)).$$

Jelikož nevíme, kterou $P \in \Pi$ máme vzít do úvahy při hodnocení rozhodovací funkce, můžeme vzít (pesimistické) kritérium odpovídající maximální možné chybě, které se můžeme dopustit

$$\max_{P \in \Pi} e_P(d) = \max_{P \in \Pi} \sum_{i=1}^2 \sum_{\mathbf{r} \in \mathbf{Q}} P(\mathbf{r}, a_i) (1 - \delta(d(\mathbf{r}), a_i)).$$

Vzhledem k tomuto kritériu je optimální rozhodovací funkce

$$d_{OPT} = \arg \min_d \max_{P \in \Pi} \sum_{i=1}^2 \sum_{\mathbf{r} \in \mathbf{Q}} P(\mathbf{r}, a_i) (1 - \delta(d(\mathbf{r}), a_i)).$$

Z tohoto vyjádření také vychází pojmenování principu “minimaxu”.

Výše uvedené použití principu minimaxu však není jediné možné. Nejprve si převedme vzorec vyjadřující střední chybu rozhodovací funkce do tvaru, který bude výhodnější

pro další úvahy.

$$\begin{aligned}
 e_P(d) &= \sum_{i=1}^2 \sum_{\mathbf{r} \in \mathbf{Q}} P(\mathbf{r}, a_i)(1 - \delta(d(\mathbf{r}), a_i)) = \\
 &= \sum_{\mathbf{r} \in \mathbf{Q}} \left[\sum_{i=1}^2 P(\mathbf{r}, a_i) - \sum_{i=1}^2 P(\mathbf{r}, a_i) \delta(d(\mathbf{r}), a_i) \right] = \\
 &= \sum_{\mathbf{r} \in \mathbf{Q}} (P(\mathbf{r}) - P(\mathbf{r}, d(\mathbf{r}))).
 \end{aligned}$$

Hledejme nyní konkrétní hodnotu rozhodovací funkce $d_{\underline{OPT}}$ v bodě \mathbf{r} tak, aby její příspěvek k výše uvedenému výrazu vyhovoval principu minimaxu. V tom případě musíme definovat

$$d_{\underline{OPT}}(\mathbf{r}) = \arg \min_{a \in \{a_1, a_2\}} \max_{P \in \Pi} (P(\mathbf{r}) - P(\mathbf{r}, d(a))).$$

Na tomto místě je vhodné upozornit, že obecně funkce $d_{\overline{OPT}}$ a $d_{\underline{OPT}}$ mohou být různé.

Příklad. Uvažujme stejně jako v minulém příkladu $\mathbf{Q} = \{0, 1\}$ a třídu Π pravděpodobnostních distribucí $\Pi = \{P_1, P_2\}$ definovaných na $\mathbf{Q} \times \{a_1, a_2\}$:

P_1	a_1	a_2
$\mathbf{r} = 0$	0.2	0.1
$\mathbf{r} = 1$	0.4	0.3

P_2	a_1	a_2
$\mathbf{r} = 0$	0.1	0.6
$\mathbf{r} = 1$	0.1	0.2

Opět musíme samozřejmě uvažovat všechny čtyři rozhodovací funkce

$$\begin{aligned}
 d_1(0) &= a_1, & d_1(1) &= a_1, \\
 d_2(0) &= a_1, & d_2(1) &= a_2, \\
 d_3(0) &= a_2, & d_3(1) &= a_1, \\
 d_4(0) &= a_2, & d_4(1) &= a_2.
 \end{aligned}$$

Nelze bayesovskou optimální rozhodovací funkci či rozhodovací funkci optimální ve smyslu maximální věrohodnosti pro obě tyto distribuce je snadné. V prvním případě přiřazujeme hodnotě $\mathbf{r} \in \mathbf{Q}$ to rozhodnutí, pro které je $P(\mathbf{r}, a_i)$ větší. Tedy bayesovsky optimální jsou funkce d_1 pro P_1 a d_4 pro P_2 .

Při hledání rozhodovacích funkcí optimálních ve smyslu maximální věrohodnosti postupujeme obdobně, ale používáme tabulky obsahující podmíněné pravděpodobnosti

$P_1(\mathbf{r} a_i)$	a_1	a_2
$\mathbf{r} = 0$	$\frac{1}{3}$	$\frac{1}{4}$
$\mathbf{r} = 1$	$\frac{2}{3}$	$\frac{3}{4}$

$P_2(\mathbf{r} a_i)$	a_1	a_2
$\mathbf{r} = 0$	$\frac{1}{2}$	$\frac{3}{4}$
$\mathbf{r} = 1$	$\frac{1}{2}$	$\frac{1}{4}$

a okamžitě dostáváme, že v tomto smyslu jsou optimální rozhodovací funkce d_2 pro P_1 a d_3 pro P_2 .

Doposud jsme uvažovali pouze optimalitu vzhledem k jedné z distribucí. Nyní se budeme zabývat tématem tohoto odstavce, t.j. budeme uvažovat situaci, kdy nevíme, která z uvažovaných distribucí je správná. Princip maximální entropie nás vede k tomu, abychom vybrali distribuci P_1 , která má entropii větší než P_2 a dále uvažovali rozhodovací funkci d_1 nebo d_2 , o kterých jsme ukázali, že jsou v jistém smyslu optimální při P_1 .

Princip minimaxu nás vede k tomu, abychom spočítali pravděpodobnosti chyb $e_{P_j}(d_i)$ pro všechny čtyři rozhodovací funkce a obě distribuce. Uvedené hodnoty jsou v následující tabulce.

d_i	$e_{P_1}(d_i)$	$e_{P_2}(d_i)$	$\max_{P \in \Pi} e_P(d_i)$
d_1	0.4	0.8	0.8
d_2	0.5	0.7	0.7
d_3	0.5	0.3	0.5
d_4	0.6	0.2	0.6

Z této tabulky vidíme, že $d_{OPT} = d_3$, neboť tato funkce minimalizuje hodnotu v posledním sloupci.

Při hledání funkce d_{OPT} hledáme hodnotu funkce d_{OPT} v každém bodě $\mathbf{r} \in \mathbf{Q}$ zvlášť. Hodnoty $(P_j(\mathbf{r}) - P_j(\mathbf{r}, a_i)) = P_j(\mathbf{r}, a_{3-i})$ dosazujeme přímo z tabulek definujících distribuce

$(P_j(0) - P_j(0, a_i))$	a_1	a_2
P_1	0.1	0.2
P_2	0.6	0.1

$(P_j(1) - P_j(1, a_i))$	a_1	a_2
P_1	0.3	0.4
P_2	0.2	0.1

To znamená, že pro jejich maximum dostáváme hodnoty

$\max_{P \in \Pi} (P(\mathbf{r}) - P(\mathbf{r}, a_i))$	a_1	a_2
$\mathbf{r} = 0$	0.6	0.2
$\mathbf{r} = 1$	0.3	0.4

a tedy v tomto případě dostáváme, že i $d_{OPT} = d_3$.

4.4 Cvičení

1. Kolik různých rozhodovacích funkcí

$$d : \times_{i=1}^N Q_i \longrightarrow \{a_1, a_2\}$$

existuje, jestliže všechny uvažované otázky Q_i jsou binární?

2. Odvoďte nutné a postačující podmínky pro to, aby konstantní rozhodovací funkce

$$d(\mathbf{r}) \equiv a_1$$

byla přípustná.

3. S využitím výsledku z minulého cvičení charakterizujte situace, kdy existuje pouze jediná přípustná rozhodovací funkce.
4. Charakterizujte situace, kdy bayesovsky optimální rozhodovací funkce je optimální i vzhledem ke kritériu maximální věrohodnosti.
5. Víme, že rozhodovací funkce d je optimální vzhledem ke kritériu maximální věrohodnosti, jestliže

$$d_{ML}(\mathbf{r}) = \begin{cases} a_1 & \text{jestliže } P(\mathbf{r}|a_1) > P(\mathbf{r}|a_2), \\ a_2 & \text{jestliže } P(\mathbf{r}|a_1) < P(\mathbf{r}|a_2). \end{cases}$$

Ukažte, že pro bayesovsky optimální rozhodovací funkci platí

$$d_B(\mathbf{r}) = \begin{cases} a_1 & \text{jestliže } P(a_1|\mathbf{r}) > P(a_2|\mathbf{r}), \\ a_2 & \text{jestliže } P(a_1|\mathbf{r}) < P(a_2|\mathbf{r}). \end{cases}$$

6. Zobecněte vzorce pro celkové riziko (či ztrátu) rozhodovací funkce pro alternativu s obecným počtem rozhodnutí.

4.5 Literatura

1. Hájek, P., Havránek, T., Jiroušek, R.: Uncertain Information Processing in Expert Systems. *CRC Press, Inc., Boca Raton, 1992.*

Kapitola 5

Skládání částečných znalostí

V této kapitole se budeme zabývat způsobem reprezentace a skládání nejistých znalostí v expertních systémech založených na pravidlech; v tak zvaných produkčních systémech. Jako příklad si vybereme systém PROSPECTOR. Vzhledem k tomu, že se budeme zabývat jejich základními teoretickými vlastnostmi, budeme si bázi znalostí tohoto systému výrazně zjednodušovat.

5.1 Pseudobayesovký přístup

Za bázi znalostí systému PROSPECTOR budeme považovat systém pravidel typu IF-THEN. Každé takové pravidlo

« IF E THEN H »

obsahuje dva jevy (či výroky), jejichž závislost je pravidlem popisována. Jev vyskytující se v podmínce (v našem příkladu jev E) budeme nazývat *antecedentem* příslušného pravidla. Obdobně druhý jev (H) budeme v dalším nazývat *sukcedentem* uvažovaného pravidla. Aby nemohlo dojít k odvozování v nekonečném cyklu, budeme předpokládat, že systém pravidel vytváří *acyklický orientovaný graf*. Tím rozumíme, že jestliže výroky vyskytující se v antecedentech i v sukcedentech uvažovaných pravidel ztotožníme s uzly grafu, do kterého doplníme hrany odpovídající pravidlům, potom tento graf je acyklický.

Pro snazší vyjadřování budeme výroky, které se vyskytují v jednotlivých pravidlech, považovat za *evidence* nebo *hypotézy*. Pravidla, evidence i hypotézy v systému PROSPECTOR mohou být zatíženy nejistotou. Při popisu a počítání s nejistotami odpovídajícími pravidlu

« **IF E THEN H** »

se vychází ze dvou základních myšlenek:

1. jedním číslem nelze dostatečně přesně popsat závislost dvou výroků E a H ,
2. podmíněná pravděpodobnost $P(E|H)$ je obvykle snáze a přesněji definována než podmíněná pravděpodobnost $P(H|E)$.

V PROSPECTORu jsou každému pravidlu « **IF E THEN H** » přiřazena dvě čísla. První z nich nazýváme *stupněm postačitelnosti* a je definováno výrazem

$$L_S(E, H) = \frac{P(E|H)}{P(E|\bar{H})}.$$

Symbolem \bar{H} označujeme *negaci* výroku (hypotézy) H . Stupeň postačitelnosti je tedy poměrem podmíněné pravděpodobnosti evidence E při dané hypotéze H ku podmíněné pravděpodobnosti evidence E při neplatnosti této hypotézy.

Druhým číslem přiřazeným každému pravidlu je *stupeň nutnosti* definovaný výrazem

$$L_N(E, H) = \frac{P(\bar{E}|H)}{P(\bar{E}|\bar{H})}.$$

Nejistotu přiřazenou jednotlivým evidencím a hypotézám budeme vyjadřovat pomocí tak zvaných *šancí*, které jsou poměrem pravděpodobností platnosti a neplatnosti příslušné evidence či hypotézy

$$o(E) = \frac{P(E)}{P(\bar{E})}$$

a

$$o(H) = \frac{P(H)}{P(\bar{H})}.$$

Nejjednodušší možná situace při odvozování může nastat tehdy, jestliže víme, zda evidence E platí či neplatí a chceme pro odvození *aposteriorní šance* $o(H|E)$ hypotézy H použít pravidlo « **IF E THEN H** », kterému jsou přiřazeny odpovídající stupně postačitelnosti $L_S(E, H)$ a nutnosti $L_N(E, H)$. V tom případě můžeme způsob počítání přímo odvodit z definic jednotlivých používaných koeficientů.

$$o(H|E) = \frac{P(H|E)}{P(\bar{H}|E)} = \frac{P(H|E)P(E)}{P(\bar{H}|E)P(E)} = \frac{P(E|H)P(H)}{P(E|\bar{H})P(\bar{H})} = L_S(E, H)o(H)$$

při platnosti evidence E a obdobně při platnosti její negace

$$o(H|\bar{E}) = \frac{P(H|\bar{E})}{P(\bar{H}|\bar{E})} = \frac{P(H|\bar{E})P(\bar{E})}{P(\bar{H}|\bar{E})P(\bar{E})} = \frac{P(\bar{E}|H)P(H)}{P(\bar{E}|\bar{H})P(\bar{H})} = L_N(E, H)o(H).$$

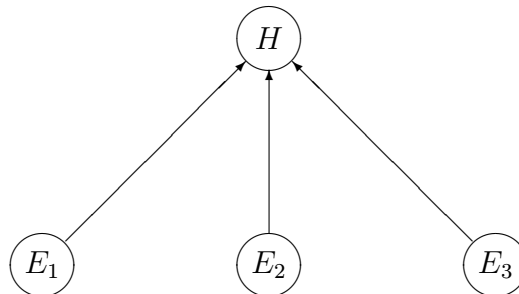
Dostali jsme tedy základní *odvozovací pravidlo*:

- Pro odvození aposteriorní šance násobíme *apriorní šanci stupněm postačitel-
nosti*, jestliže evidence obsažená v antecedentu pravidla **platí**.
- Jestliže evidence **neplatí**, násobíme apriorní šanci **stupněm nutnosti**.

Začněme se nyní zabývat situací, kdy jedna hypotéza se vyskytuje v sukcedentu několika odvozovacích pravidel, například

<< **IF** E_1 **THEN** H >>,
 << **IF** E_2 **THEN** H >>,
 << **IF** E_3 **THEN** H >>.

V grafickém vyjádření to znamená, že v příslušném grafu existují (v našem případě tři) uzly, ze kterých vedou hrany do společného následníka.



V PROSPECTORu se i v tomto případě používá výše uvedené odvozovací pravidlo. Předpokládejme například, že evidence E_1 a E_3 platí, zatímco evidence E_2 neplatí a

spočítejme, co použitím tohoto pravidla obdržíme:

$$o(H)L_S(E_1, H)L_N(E_2, H)L_S(E_3, H) = \frac{P(E_1|H)P(\bar{E}_2|H)P(E_3|H)P(H)}{P(E_1|\bar{H})P(\bar{E}_2|\bar{H})P(E_3|\bar{H})P(\bar{H})}.$$

My bychom však rádi dostali aposteriorní šanci, t.j.

$$o(H|E_1, \bar{E}_2, E_3) = \frac{P(H|E_1, \bar{E}_2, E_3)P(E_1, \bar{E}_2, E_3)}{P(\bar{H}|E_1, \bar{E}_2, E_3)P(E_1, \bar{E}_2, E_3)} = \frac{P(E_1, \bar{E}_2, E_3|H)P(H)}{P(E_1, \bar{E}_2, E_3|\bar{H})P(\bar{H})}.$$

Ovšem

$$P(E_1|H)P(\bar{E}_2|H)P(E_3|H) = P(E_1, \bar{E}_2, E_3|H)$$

a

$$P(E_1|\bar{H})P(\bar{E}_2|\bar{H})P(E_3|\bar{H}) = P(E_1, \bar{E}_2, E_3|\bar{H})$$

právě tehdy, jestliže evidence E_1, E_2, E_3 jsou vzájemně podmíněně nezávislé při hypotéze H . Odvodili jsme tedy, že PROSPECTORovské odvozovací pravidlo je *pravděpodobnostně korektní* pouze tehdy, jestliže evidence přiřazené uzlům vedoucím do stejného následníka jsou *podmíněně nezávislé* při dané hypotéze přiřazené společnému následníku. Proto se také uvedený způsob odvozování nazývá *pseudobayesovský*.

Abychom v dalším rozlišili šance teoretické od šancí počítaných pomocí pseudobayesovského přístupu používaného v PROSPECTORu, budeme tyto druhé šance označovat symbolem \hat{o} .

5.2 Metoda korekcí

Uvažujme modelovou situaci při hodu hrací kostkou. Definujme si dva jevy jako evidence

$$\begin{aligned} E_1 & \text{ výsledek je menší než 4,} \\ E_2 & \text{ výsledek je lichý,} \end{aligned}$$

a dva jevy jako hypotézy

$$\begin{aligned} H_1 & \text{ výsledek je 5,} \\ H_2 & \text{ výsledek je prvočíslo}^1. \end{aligned}$$

PROSPECTORovskou znalostní bázi vytvoříme ze čtyř pravidel:

«**IF** E_1 **THEN** H_1 »,

«**IF** E_2 **THEN** H_1 »,

«**IF** E_1 **THEN** H_2 »,

«**IF** E_2 **THEN** H_2 ».

pro která spočteme příslušné stupně postačitelnosti a nutnosti:

$$L_S(E_1, H_1) = \frac{P(E_1|H_1)}{P(E_1|\overline{H_1})} = \frac{0}{\frac{2}{5}} = 0,$$

$$L_S(E_2, H_1) = \frac{P(E_2|H_1)}{P(E_2|\overline{H_1})} = \frac{1}{\frac{2}{5}} = \frac{5}{2},$$

$$L_S(E_1, H_2) = \frac{P(E_1|H_2)}{P(E_1|\overline{H_2})} = \frac{\frac{2}{3}}{\frac{1}{3}} = 2,$$

$$L_S(E_2, H_2) = \frac{P(E_2|H_2)}{P(E_2|\overline{H_2})} = \frac{\frac{2}{3}}{\frac{1}{3}} = 2,$$

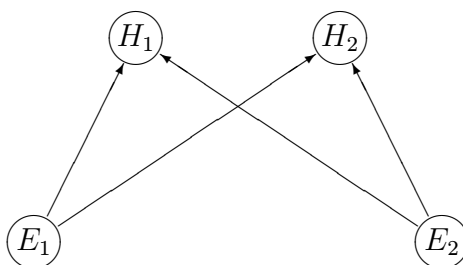
$$L_N(E_1, H_1) = \frac{P(\overline{E_1}|H_1)}{P(\overline{E_1}|\overline{H_1})} = \frac{1}{\frac{2}{5}} = \frac{5}{2},$$

$$L_N(E_2, H_1) = \frac{P(\overline{E_2}|H_1)}{P(\overline{E_2}|\overline{H_1})} = \frac{0}{\frac{2}{5}} = 0,$$

$$L_N(E_1, H_2) = \frac{P(\overline{E_1}|H_2)}{P(\overline{E_1}|\overline{H_2})} = \frac{\frac{1}{3}}{\frac{2}{3}} = \frac{1}{2},$$

$$L_N(E_2, H_2) = \frac{P(\overline{E_2}|H_2)}{P(\overline{E_2}|\overline{H_2})} = \frac{\frac{1}{3}}{\frac{2}{3}} = \frac{1}{2},$$

Nadefinovali jsme si tedy PROSPECTORovskou síť odpovídající následující situaci.



¹Pro jistotu připomínáme, že na hrací kostce jsou prvočísla: 2, 3, 5.

Spočtěme si, jaké dostaneme aposteriorní šance daných hypotéz v případě, že známe zda nastává či nenastává jedna z evidencí. Pro evidenci E_1 dostáváme

$$\hat{o}(H_1|E_1) = o(H_1) \cdot L_S(E_1, H_1) = \frac{1}{5} \cdot 0 = 0,$$

$$\hat{o}(H_1|\bar{E}_1) = o(H_1) \cdot L_N(E_1, H_1) = \frac{1}{5} \cdot \frac{5}{2} = \frac{1}{2},$$

$$\hat{o}(H_2|E_1) = o(H_2) \cdot L_S(E_1, H_2) = 1 \cdot 2 = 2,$$

$$\hat{o}(H_2|\bar{E}_1) = o(H_2) \cdot L_N(E_1, H_2) = 1 \cdot \frac{1}{2} = \frac{1}{2},$$

neboť

$$o(H_1) = \frac{P(H_1)}{P(\bar{H}_1)} = \frac{1/6}{5/6} = \frac{1}{5}, \quad o(H_2) = \frac{P(H_2)}{P(\bar{H}_2)} = \frac{1/2}{1/2} = 1.$$

Obdobně pro evidenci E_2 platí

$$\hat{o}(H_1|E_2) = o(H_1) \cdot L_S(E_2, H_1) = \frac{1}{5} \cdot \frac{5}{2} = \frac{1}{2},$$

$$\hat{o}(H_1|\bar{E}_2) = o(H_1) \cdot L_N(E_2, H_1) = \frac{1}{5} \cdot 0 = 0,$$

$$\hat{o}(H_2|E_2) = o(H_2) \cdot L_S(E_2, H_2) = 1 \cdot 2 = 2,$$

$$\hat{o}(H_2|\bar{E}_2) = o(H_2) \cdot L_N(E_2, H_2) = 1 \cdot \frac{1}{2} = \frac{1}{2}.$$

Tyto hodnoty samozřejmě odpovídají teoretickým šancím

$$o(H_i|E_j) = \frac{P(H_i|E_j)}{P(\bar{H}_i|E_j)},$$

což si může čtenář jednoduchým výpočtem ověřit.

Ovšem zcela jiná situace nastává, jestliže budeme uvažovat platnost (či neplatnost) dvou evidencí současně. V dalším se budeme podrobněji zabývat dvěma příklady. Jednak budeme uvažovat situaci, kdy obě evidences platí současně, tedy $E_1 \& E_2$, a dále pak situací, kdy E_1 neplatí a současně E_2 platí, tedy $\bar{E}_1 \& E_2$. Pro tyto situace spočítáme snadno teoretické šance:

$$o(H_2|E_1, E_2) = \frac{P(H_2|E_1, E_2)}{P(\bar{H}_2|E_1, E_2)} = \frac{1}{2} / \frac{1}{2} = 1,$$

$$o(H_1|\bar{E}_1, E_2) = \frac{P(H_1|\bar{E}_1, E_2)}{P(\bar{H}_1|\bar{E}_1, E_2)} = 1/0 = +\infty.$$

Nicméně PROSPECTORovské zacházení s nejistotou dá hodnoty

$$\hat{o}(H_2|E_1, E_2) = o(H_2) \cdot L_S(E_1, H_2) \cdot L_S(E_2, H_2) = 1 \cdot 2 \cdot 2 = 4,$$

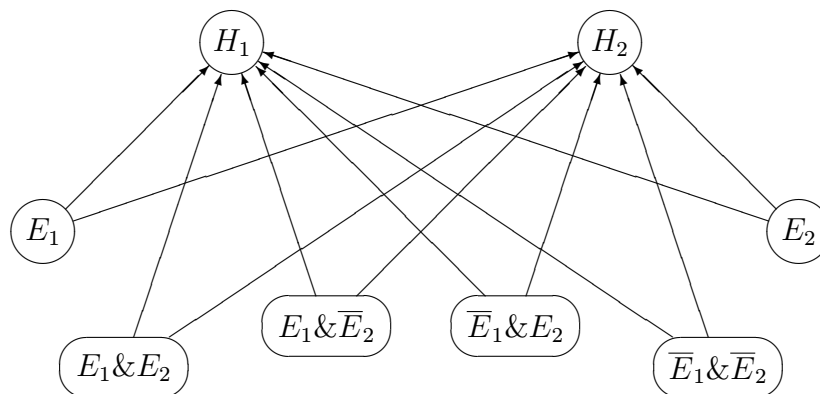
a

$$\hat{o}(H_1|\bar{E}_1, E_2) = o(H_1) \cdot L_N(E_1, H_1) \cdot L_S(E_2, H_1) = \frac{1}{5} \cdot \frac{5}{2} \cdot \frac{5}{2} = \frac{5}{4},$$

což odpovídá faktu, že evidences E_1 a E_2 nejsou podmíněně nezávislé při ani jedné z hypotéz H_1 či H_2 .

Na tomto příkladu si ještě ukážeme způsob, jak lze tento nedostatek napravit. Metodu navrhl Petr Hájek pro práci se závislými evidencemi v obecných pravidlově orientovaných expertních systémech. Samozřejmě, že postup je použitelný pro libovolný počet pravidel se stejným sukcedentem. My jej zde předvedeme pro námi uvažované dvě evidences a zobecnění postupu pro více evidencí ponecháme na čtenáři.

Postup, jak jsme si jej popsali, selhává, jestliže víme o obou evidencích, zda nastávají, či ne. Doplňme si proto do uvažované báze znalostí čtyři doplňková pravidla (do grafu čtyři uzly) podle následujícího schématu.



Jestliže nyní známe konkrétní stav platnosti obou evidencí E_1 a E_2 , to je víme o obou evidencích, zda nastávají či ne, potom v každém případě platí právě jeden ze čtyř nově doplněných uzlů. Budeme-li definovat stupně nutnosti doplněných uzlů

$$L_N(E_1 \& E_2, H_i) = L_N(E_1 \& \bar{E}_2, H_i) = L_N(\bar{E}_1 \& E_2, H_i) = L_N(\bar{E}_1 \& \bar{E}_2, H_i) = 1,$$

pak se při PROSPECTORovském výpočtu aposteriorních šancí uplatní vlastně pouze stupně postačitelnosti nově doplněných uzlů, které používáme při platnosti odpovídající (v našem případě složené) evidence. Definujme si tedy

$$L_S(E_1 \& E_2, H_i) = \frac{1}{L_S(E_1, H_i)L_S(E_2, H_i)} \cdot \frac{P(E_1 \& E_2 | H_i)}{P(E_1 \& E_2 | \bar{H}_i)}.$$

Potom při použití PROSPECTORovského skládání nejistot budeme při znalosti faktu, že obě dvě evidencí platí, počítat

$$\begin{aligned} o(H_i | E_1 \& E_2) &= o(H_i)L_S(E_1, H_i)L_S(E_2, H_i)L_S(E_1 \& E_2, H_i) \\ &\quad L_N(E_1 \& \bar{E}_2, H_i)L_N(\bar{E}_1 \& E_2, H_i)L_N(\bar{E}_1 \& \bar{E}_2, H_i) = \\ &= o(H_i)L_S(E_1, H_i)L_S(E_2, H_i) \\ &\quad \frac{1}{L_S(E_1, H_i)L_S(E_2, H_i)} \cdot \frac{P(E_1 \& E_2 | H_i)}{P(E_1 \& E_2 | \bar{H}_i)} \cdot 1 \cdot 1 \cdot 1 = \\ &= \frac{P(E_1 \& E_2, H_i)}{P(E_1 \& E_2, \bar{H}_i)} = \frac{P(H_i | E_1 \& E_2)}{P(\bar{H}_i | E_1 \& E_2)} = o(H_i | E_1 \& E_2), \end{aligned}$$

což je přesně to, co jsme požadovali.

Proto i pro ostatní doplněné uzly dodefinujeme stupně postačitelnosti

$$L_S(E_1 \& \bar{E}_2, H_i) = \frac{1}{L_S(E_1, H_i)L_S(\bar{E}_2, H_i)} \cdot \frac{P(E_1 \& \bar{E}_2 | H_i)}{P(E_1 \& \bar{E}_2 | \bar{H}_i)},$$

$$L_S(\bar{E}_1 \& E_2, H_i) = \frac{1}{L_S(\bar{E}_1, H_i) L_S(E_2, H_i)} \cdot \frac{P(\bar{E}_1 \& E_2 | H_i)}{P(\bar{E}_1 \& E_2 | \bar{H}_i)},$$

$$L_S(\bar{E}_1 \& \bar{E}_2, H_i) = \frac{1}{L_S(\bar{E}_1, H_i) L_S(\bar{E}_2, H_i)} \cdot \frac{P(\bar{E}_1 \& \bar{E}_2 | H_i)}{P(\bar{E}_1 \& \bar{E}_2 | \bar{H}_i)}.$$

5.3 Cvičení

1. Dokažte, že pro libovolné pravidlo

$$\ll \mathbf{IF } E \mathbf{ THEN } H \gg$$

platí

$$L_N = \frac{1 - L_S P(E | \bar{H})}{1 - P(E | \bar{H})}.$$

2. Spočtěte stupně postačitelnosti pro doplněná pravidla

$$\ll \mathbf{IF } E_1 \& E_2 \mathbf{ THEN } H_1 \gg,$$

$$\ll \mathbf{IF } E_1 \& \bar{E}_2 \mathbf{ THEN } H_1 \gg,$$

$$\ll \mathbf{IF } \bar{E}_1 \& E_2 \mathbf{ THEN } H_1 \gg,$$

$$\ll \mathbf{IF } \bar{E}_1 \& \bar{E}_2 \mathbf{ THEN } H_1 \gg,$$

$$\ll \mathbf{IF } E_1 \& E_2 \mathbf{ THEN } H_2 \gg,$$

$$\ll \mathbf{IF } E_1 \& \bar{E}_2 \mathbf{ THEN } H_2 \gg,$$

$$\ll \mathbf{IF } \bar{E}_1 \& E_2 \mathbf{ THEN } H_2 \gg,$$

$$\ll \mathbf{IF } \bar{E}_1 \& \bar{E}_2 \mathbf{ THEN } H_2 \gg,$$

z výše uvedeného příkladu házení hrací kostkou.

3. Navrhněte zobecnění Hájkovy metody pro případ, že v datové bázi jsou tři pravidla se stejným sukcedentem

« IF E_1 THEN H »,

« IF E_2 THEN H »,

« IF E_3 THEN H ».

Kolik doplňkových pravidel (uzlů) potřebujete v tomto případě?

5.4 Literatura

1. Hájek, P.: Combining functions for certainty factors in consulting systems. *Int. J. Man-Machine Studies*, 22(1985), pp59-67.
2. Hájek, P., Havránek, T., Jiroušek, R.: Uncertain Information Processing in Expert Systems. *CRC Press, Inc., Boca Raton, 1992.*
3. Lopez de Mántaras, R.: Approximate reasoning models. *Ellis Horwood, 1990.*

Kapitola 6

Bayesovské sítě

Abychom nemuseli složitě odstraňovat nedostatky PROSPECTORovské sítě pomocí v minulé kapitole uvedené Hájkovy metody, budeme se nadále věnovat bayesovským sítím, které uvedenými nedostatky netrpí. Jedná se o dosti obecnou metodu reprezentace znalostí, která umožňuje zavádět poměrně složité závislosti mezi jednotlivé evidence a hypotézy. Její obecnost nás vede také k tomu, abychom nadále nerozlišovali uvažované proměnné na evidence a hypotézy. Budeme je všechny nadále nazývat náhodnými veličinami.

Bayesovská síť (v literatuře také nazývaná *kauzální* nebo též *influenční diagram*) je tvořena acyklickým¹ orientovaným grafem $G = (V, \vec{E})$, jehož uzlům jsou přiřazeny náhodné veličiny (přiřazení je vzájemně jednoznačné) a systémem podmíněných pravděpodobnostních distribucí

$$\{P(X_i | (X_j)_{j \in pa(i)})\}_{i \in V}.$$

Pro každý uzel grafu (neboli pro každou uvažovanou náhodnou veličinu) je tedy zadána podmíněná pravděpodobnostní distribuce. Poznamenejme, že pro *zdrojové* uzly (uzly, které nemají rodičů) jsou tyto distribuce vlastně nepodmíněné.

¹Připomeňme si, že orientovaný graf $G = (V, \vec{E})$ je *acyklický*, jestliže v něm neexistuje *orientovaný cyklus*, t.j. posloupnost i_1, \dots, i_n uzlů pro které $i_n = i_1$ a $i_m \in pa(i_{m+1})$ pro všechna $m = 1, \dots, n-1$. Symbolem $pa(i)$, pro uzel i orientovaného grafu $G = (V, \vec{E})$, značíme jako obvykle množinu jeho *rodičů*, t.j. množinu uzlů, ze kterých vede do i hrana:

$$pa(i) = \{j \in V : (j \rightarrow i) \in \vec{E}\}.$$

6.1 Distribuce reprezentovaná bayesovskou sítí

Bayesovskou sítí jsou zadány znalosti dvojího druhu. Znalosti kvantitativní, popisující pomocí podmíněných pravděpodobnostních distribucí závislostní vztahy mezi jednotlivými veličinami, a znalosti kvalitativní, které budeme nazývat též znalostmi *strukturálními*. Ty jsou zakódovány pomocí zadaného acyklického orientovaného grafu a vyjadřují, které závislosti (či spíše, které nezávislosti) musíme při práci s bayesovskou sítí respektovat.

Bayesovské sítí totiž nepřičítáme libovolnou distribuci $P(X_i)_{i \in V}$, jejíž příslušné podmíněné pravděpodobnosti se shodují se zadanými, ale pouze tu jedinou, jejíž závislostní struktura odpovídá zadanému grafu. Abychom si mohli jednoznačně popsat, o kterou distribuci se jedná, musíme si zde nejdříve definovat takzvaný *předpoklad nezávislosti*.

Nalezneme nejprve nějaké očíslování uzlů uvažovaného acyklického grafu $G = (V, \vec{E})$, neboli vzájemně jednoznačné přiřazení uzlů grafu a přirozených čísel

$$no : V \longrightarrow \{1, \dots, |V|\}$$

takové, že rodiče každého uzlu mají přiřazeno nižší číslo, než jejich potomek²

$$i \in pa(j) \implies no(i) < no(j).$$

Každé takové očíslování může generovat systém podmíněných nezávislostí

$$\forall i \in V \quad (X_i \perp (X_j)_{j \in (\{k \in V : no(k) < no(i)\} - pa(i))} | (X_j)_{j \in pa(i)}).$$

V souvislosti s bayesovskou sítí se budeme zajímat pouze o ty distribuce, které takový systém nezávislostí splňují. Proto je nesmírně důležité následující tvrzení, které ukazuje, že z faktického hlediska nezáleží na vybraném očíslování uzlů.

Tvrzení 4. Uvažujme bayesovskou síť, která je zadána grafem $G = (V, \vec{E})$ a systémem podmíněných pravděpodobnostních distribucí

$$\{P(X_i | (X_j)_{j \in pa(i)})\}_{i \in V},$$

a libovolné očíslování

$$no : V \longrightarrow \{1, \dots, |V|\}$$

všech uzlů grafu G takové, že

$$i \in pa(j) \implies no(i) < no(j).$$

²Poznamenejme, že takovéto očíslování vždy existuje. Viz též cíčení 1. v této kapitole.

Potom existuje právě jedna pravděpodobnostní distribuce $R((X_i)_{i \in V})$, pro kterou

$$R(X_i | (X_j)_{j \in pa(i)}) = P(X_i | (X_j)_{j \in pa(i)})$$

a

$$X_i \perp_R (X_j)_{j \in (\{k \in V: no(k) < no(i)\} - pa(i))} | (X_j)_{j \in pa(i)}$$

pro všechna $i \in V$. Navíc, tuto distribuci je možno vypočítat podle výrazu

$$R((X_i)_{i \in V}) = \prod_{i \in V} P(X_i | (X_j)_{j \in pa(i)}).$$

Poznámka. Nezávislost na volbě uspořádání (očíslování) uzlů vyplývá okamžitě ze vzorce pro výpočet distribuce R , který na uspořádání nezávisí.

Důkaz. Dokažme tvrzení matematickou indukcí přes počet uzlů (veličin) uvažované bayesovské sítě.

Je-li $|V| = 1$, dostáváme degenerovaný případ, kdy bayesovská síť je zadána jedinou distribucí $P(X)$ a tvrzení je triviální.

Uvažujme nyní obecnou bayesovskou síť s n veličinami a předpokládejme, v souladu s indukčním předpokladem, že tvrzení platí pro všechny bayesovské sítě s nižším počtem veličin.

Uvažujme libovolné očíslování uzlů

$$no : V \longrightarrow \{1, \dots, |V|\}$$

vyhovující požadované podmínce

$$i \in pa(j) \implies no(i) < no(j).$$

Nechť $k \in V$ je ten uzel, kterému je přiřazeno číslo $no(k) = n = |V|$. Je zřejmé, že graf

$$G' = (V - \{k\}, \vec{E} - \{(i \rightarrow k)\}_{i \in pa(k)})$$

spolu se systémem distribucí

$$\{P(X_i | (X_j)_{j \in pa(i)})\}_{i \in V - \{k\}}$$

tvoří bayesovskou síť, na kterou se vztahuje indukční předpoklad. Proto existuje jediná distribuce (označme si ji R')

$$R'((X_i)_{i \in V - \{k\}}) = \prod_{i \in V - \{k\}} P(X_i | (X_j)_{j \in pa(i)}),$$

pro kterou platí

$$R'(X_i|(X_j)_{j \in pa(i)}) = P(X_i|(X_j)_{j \in pa(i)})$$

a

$$X_i \perp_{R'} (X_j)_{j \in (\{l \in V : no(l) < no(i)\} - pa(i)) | (X_j)_{j \in pa(i)}}$$

pro všechna $i \in V - \{k\}$. Podle indukčního předpokladu je $R'((X_i)_{i \in V - \{k\}})$ jediná, která splňuje tyto podmínky a proto musí být marginálem námi hledané distribuce $R((X_i)_{i \in V})$. Vzorec

$$R((X_i)_{i \in V}) = R(X_k|(X_i)_{i \in V - \{k\}})R((X_i)_{i \in V - \{k\}})$$

platí pro libovolnou distribuci a tedy i pro námi uvažovanou distribuci R . Jelikož $\{l \in V : no(l) < no(k)\} = V - \{k\}$, tak z požadavku

$$X_k \perp_R (X_j)_{j \in (\{l \in V : no(l) < no(k)\} - pa(k)) | (X_j)_{j \in pa(k)}}$$

dostáváme

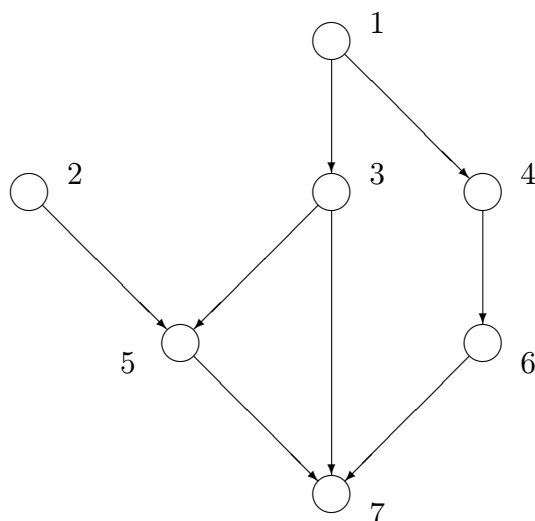
$$R(X_k|(X_j)_{j \in V - \{k\}}) = R(X_k|(X_j)_{j \in pa(k)}),$$

a tedy

$$\begin{aligned} R((X_i)_{i \in V}) &= R(X_k|(X_i)_{i \in V - \{k\}})R((X_i)_{i \in V - \{k\}}) = \\ &= R(X_k|(X_i)_{i \in pa(k)})R'((X_i)_{i \in V - \{k\}}) = \\ &= P(X_k|(X_i)_{i \in pa(k)}) \prod_{i \in V - \{k\}} P(X_i|(X_j)_{j \in pa(i)}) = \\ &= \prod_{i \in V} P(X_i|(X_j)_{j \in pa(i)}), \end{aligned}$$

což jsme chtěli dokázat. Jednoznačnost této distribuce plyne z jednoznačnosti prováděných úprav. \square

Příklad. Ilustrujme si definici bayesovské sítě a uvedené vlastnosti na jednoduchém příkladu se sedmi binárními náhodnými veličinami (X_1, \dots, X_7) . Graf uvažované bayesovské sítě je na následujícím obrázku.



Odpovídající systém pravděpodobnostních distribucí je zadán v následující tabulce, která obsahuje pro všechny náhodné veličiny $P(X_i = 1 | (X_j)_{j \in pa(i)})$. Vzhledem k tomu, že uvažujeme binární veličiny, tak hodnoty

$$P(X_i = 0 | (X_j)_{j \in pa(i)})$$

můžeme snadno a jednoznačně dopočítat.

Tabulka podmíněných pravděpodobnostních distribucí.

$P(X_1 = 1)$	0.5
$P(X_2 = 1)$	0.5
$P(X_3 = 1 X_1 = 0)$	0.25
$P(X_3 = 1 X_1 = 1)$	0.75
$P(X_4 = 1 X_1 = 0)$	0.35
$P(X_4 = 1 X_1 = 1)$	0.85
$P(X_5 = 1 X_2 = 0, X_3 = 0)$	1
$P(X_5 = 1 X_2 = 0, X_3 = 1)$	0
$P(X_5 = 1 X_2 = 1, X_3 = 0)$	0
$P(X_5 = 1 X_2 = 1, X_3 = 1)$	1
$P(X_6 = 1 X_4 = 0)$	0.25
$P(X_6 = 1 X_4 = 1)$	0.5
$P(X_7 = 1 X_3 = 0, X_5 = 0, X_6 = 0)$	0.3
$P(X_7 = 1 X_3 = 0, X_5 = 0, X_6 = 1)$	0.15
$P(X_7 = 1 X_3 = 0, X_5 = 1, X_6 = 0)$	0
$P(X_7 = 1 X_3 = 0, X_5 = 1, X_6 = 1)$	0.3
$P(X_7 = 1 X_3 = 1, X_5 = 0, X_6 = 0)$	0.15
$P(X_7 = 1 X_3 = 1, X_5 = 0, X_6 = 1)$	0.55
$P(X_7 = 1 X_3 = 1, X_5 = 1, X_6 = 0)$	0.6
$P(X_7 = 1 X_3 = 1, X_5 = 1, X_6 = 1)$	0.1

Podle předchozího tvrzení, zadané bayesovské síti odpovídá jediná pravděpodobnostní distribuce definovaná výrazem

$$P(X_1, X_2, \dots, X_7) = \\ = P(X_1)P(X_2)P(X_3|X_1)P(X_4|X_1)P(X_5|X_2, X_3)P(X_6|X_4)P(X_7|X_3, X_5, X_6).$$

Tato distribuce je jediná, přestože uvažovanému grafu odpovídá 26 různých uspořádání uzlů odpovídajících požadavku, aby rodiče předcházeli své potomky. Všech těchto 26 uspořádání je obsaženo v následující tabulce:

Možná uspořádání uzlů grafu	
1 2 3 4 5 6 7	1 3 2 4 6 5 7
2 1 3 4 5 6 7	1 2 4 3 6 5 7
1 3 2 4 5 6 7	1 2 4 6 3 5 7
1 2 4 3 5 6 7	1 4 2 3 6 5 7
1 4 2 3 5 6 7	1 4 2 6 3 5 7
2 1 4 3 5 6 7	1 4 6 2 3 5 7
1 3 4 2 5 6 7	2 1 4 3 6 5 7
1 4 3 2 5 6 7	2 1 4 6 3 5 7
1 2 3 5 4 6 7	1 3 4 2 6 5 7
2 1 3 5 4 6 7	1 3 4 6 2 5 7
1 3 2 5 4 6 7	1 4 3 2 6 5 7
1 2 3 4 6 5 7	1 4 3 6 2 5 7
2 1 3 4 6 5 7	1 4 6 3 2 5 7

Z každého uspořádání můžeme sestavit systém požadavků na podmíněné nezávislosti, které výsledná distribuce splňuje. Nebudeme si zde vypisovat všech 26 systémů, ve kterých se samozřejmě většina předpokladů stále opakuje. Pouze jako příklad si uveďme požadavky odpovídající prvnímu a poslednímu uspořádání:

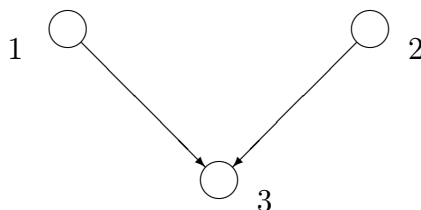
Podmíněné nezávislosti pro uspořádání

1 2 3 4 5 6 7	1 4 6 3 2 5 7
$X_2 \perp X_1$	$X_6 \perp X_1 X_4$
$X_3 \perp X_2 X_1$	$X_3 \perp (X_4, X_6) X_1$
$X_4 \perp (X_2, X_3) X_1$	$X_2 \perp (X_1, X_3, X_4, X_6)$
$X_5 \perp (X_1, X_4) (X_2, X_3)$	$X_5 \perp (X_1, X_4, X_6) (X_2, X_3)$
$X_6 \perp (X_1, X_2, X_3, X_5) X_4$	$X_7 \perp (X_1, X_2, X_4) (X_3, X_5, X_6)$
$X_7 \perp (X_1, X_2, X_4) (X_3, X_5, X_6)$	

Poznámka. Přestože to není na první pohled patrné, lze dokázat, že tyto systémy nezávislostí jsou ekvivalentní v tom smyslu, že každá distribuce splňující jeden z těchto systémů musí splňovat také všechny ostatní.

Poznámka. Uvedený předpoklad podmíněné nezávislosti je při práci s bayesovskými sítěmi nezbytný. V následujícím příkladu si ukážeme, že jej nelze nahradit principem maximální entropie.

Příklad. Uvažujme bayesovskou síť



se třemi náhodnými veličinami X_1, X_2, X_3 a tedy i třemi podmíněnými pravděpodobnostními distribucemi $P_1(X_1)$, $P_2(X_2)$ and $P_3(X_3|X_1, X_2)$ jejichž hodnoty jsou obsaženy v následující tabulce.

$P_1(X_1 = 1) = P_1(X_1 = 0) = P_2(X_2 = 1) = P_2(X_2 = 0) = 0.5$
$P_3(X_3 = 1 X_1 = 0, X_2 = 0) = P_3(X_3 = 1 X_1 = 1, X_2 = 1) = 0.5$
$P_3(X_3 = 0 X_1 = 0, X_2 = 0) = P_3(X_3 = 0 X_1 = 1, X_2 = 1) = 0.5$
$P_3(X_3 = 1 X_1 = 0, X_2 = 1) = P_3(X_3 = 1 X_1 = 1, X_2 = 0) = \varepsilon$
$P_3(X_3 = 0 X_1 = 0, X_2 = 1) = P_3(X_3 = 0 X_1 = 1, X_2 = 0) = 1 - \varepsilon$

Čtenář sám snadno ověří, že všechny třírozměrné pravděpodobnostní distribuce $P(X_1, X_2, X_3)$, které jsou konzistentní se zadanými distribucemi P_1, P_2, P_3 , jsou právě ty, pro které existuje $\alpha \in \langle 0, 0.5 \rangle$ takové, že $P(X_1, X_2, X_3) = P_\alpha(X_1, X_2, X_3)$. Hodnoty distribucí P_α jsou obsaženy v následující tabulce.

$P_\alpha(X_1, X_2, X_3)$	$X_3 = 1$	$X_3 = 0$	Σ
$X_1 = 1, X_2 = 1$	$(\frac{1}{2} - \alpha)/2$	$(\frac{1}{2} - \alpha)/2$	$\frac{1}{2} - \alpha$
$X_1 = 1, X_2 = 0$	$\alpha\varepsilon$	$\alpha(1 - \varepsilon)$	α
$X_1 = 0, X_2 = 1$	$\alpha\varepsilon$	$\alpha(1 - \varepsilon)$	α
$X_1 = 0, X_2 = 0$	$(\frac{1}{2} - \alpha)/2$	$(\frac{1}{2} - \alpha)/2$	$\frac{1}{2} - \alpha$
Σ	$\frac{1}{2}$	$\frac{1}{2}$	1

Distribuce reprezentovaná odpovídající bayesovskou sítí

$$P_1(X_1)P_2(X_2)P_3(X_3|X_1, X_2)$$

odpovídá tedy distribuci P_α pro $\alpha = \frac{1}{4}$ nezávisle na hodnotě parametru ε .

Shannonovská entropie P_α

$$H(P_\alpha) = (2\alpha - 1) \log \left[\left(\frac{1}{2} - \alpha \right) / 2 \right] - 2\alpha\varepsilon \log \alpha\varepsilon - 2\alpha(1 - \varepsilon) \log(\alpha(1 - \varepsilon))$$

je tabelována v níže uvedené tabulce. Z ní okamžitě vidíme, že $P_{\frac{1}{4}}$ maximalizuje Shannonovu entropii pro $\varepsilon = \frac{1}{2}$, ale již ne pro $\varepsilon > \frac{1}{2}$. Proto předpoklad nezávislosti, který uvažujeme při práci s bayesovskými sítěmi, nemůže být nahrazen principem maximální entropie.

Hodnoty $H(P_\alpha)$ v závislosti na ε

α	$\varepsilon = 0.5$	$\varepsilon = 0.7$	$\varepsilon = 0.8$	$\varepsilon = 0.9$	$\varepsilon = 1.0$
.000	1.3863	1.3863	1.3863	1.3863	1.3863
.090	1.8577	1.8429	1.8230	1.7914	1.7329
.095	1.8725	1.8569	1.8359	1.8026	1.7408
.100	1.8867	1.8702	1.8481	1.8131	1.7481
.160	2.0132	1.9868	1.9519	1.8954	1.7914
.165	2.0205	1.9933	1.9569	1.8990	1.7917
.170	2.0273	1.9994	1.9618	1.9022	1.7917
.200	2.0593	2.0264	1.9822	1.9121	1.7820
.205	2.0632	2.0294	1.9841	1.9122	1.7790
.210	2.0666	2.0320	1.9856	1.9120	1.7755
.230	2.0762	2.0384	1.9876	1.9069	1.7574
.235	2.0776	2.0390	1.9870	1.9046	1.7519
.240	2.0786	2.0391	1.9861	1.9020	1.7459
.245	2.0792	2.0389	1.9848	1.8989	1.7396
.250	2.0794	2.0383	1.9831	1.8954	1.7329
.255	2.0792	2.0373	1.9809	1.8915	1.7257

6.2 Odvozování metodou postupných modifikací

Nyní se začneme zabývat způsobem, jakým jsou znalosti zaznamenané v bayesovské síti využívány k inferenci. Z teoretického hlediska je jejich použití přímočaré. Každá bayesovská síť definuje jednoznačně pravděpodobnostní distribuci a tedy inference bude odpovídat výpočtu podmíněné distribuce jedné veličiny (odpovídající hypotéze) při zadaných hodnotách několika jiných veličin. Z výpočetního pohledu však zde nastává velký problém. Předpokládáme-li, že uvažovaná bayesovská síť má 100 uzlů (náhodných veličin) a počítáme-li podmíněnou distribuci jedné veličiny při znalosti hodnot devíti jiných veličin, musíme před výpočtem požadované podmíněné distribuce vypočítat nejprve 10-rozměrnou marginální distribuci. Výpočet každé jediné pravděpodobnosti tohoto marginálu však představuje součet alespoň 2^{90} čísel (v případě, že veličiny nejsou binární, je počet sčítaných pravděpodobností ještě podstatně vyšší).

Je tedy zřejmé, že tento přímočarý postup k cíli nevede. Musíme zde využít znalosti toho, že uvažovaná distribuce není zcela obecná, ale že je zadána formou bayesovské sítě. Tato znalost nám pomůže nalézt efektivní postup pro výpočet libovolné podmíněné marginální distribuce.

V tomto odstavci se budeme zabývat metodou *postupných modifikací bayesovské sítě* navrženou R. D. Shachterem. Ta je založena na myšlence postupných transformací vý-

chozí bayesovské sítě na jiné tak, že se reprezentovaná distribuce buď nezmění (v případě, že se počet uzlů nezmění), nebo bude marginálem k distribuci původní (jestliže se počet uzlů zmenší). Takovými postupnými změnami budeme hledat bayesovskou síť, která bude obsahovat jenom předem zadané uzly a bude reprezentovat požadovanou marginální distribuci.

Abychom dosáhli tohoto cíle, budeme postupně aplikovat dvě jednoduchá pravidla:

- **Vypuštění uzlu.** Jestliže z některého uzlu bayesovské sítě nevede žádná hrana, potom může být tento uzel vypuštěn (včetně příslušné podmíněné pravděpodobnostní distribuce) a distribuce reprezentovaná výslednou bayesovskou sítí (to je sítí, jejíž graf obsahuje o jeden uzel méně) bude marginálem k distribuci reprezentované sítí původní.
- **Otočení hrany.** Jestliže v uvažované bayesovské sítí existuje hrana $(k \rightarrow l)$ taková, že z uzlu k nevede do uzlu l cesta různá od uvažované hrany, potom provedení transformace popsané v následujících třech krocích nezmění reprezentovanou distribuci.

1. Přidejte nové hrany ke grafu bayesovské sítě tak, aby uzly k a l “zdědily” navzájem své rodiče:

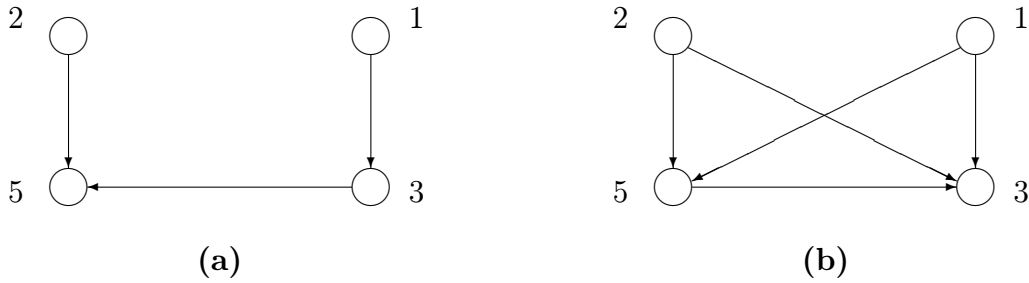
$$\{(i \rightarrow l) : i \in pa(k)\} \text{ and } \{(i \rightarrow k) : i \in (pa(l) - \{k\})\}.$$

2. Otočte orientaci hrany $(k \rightarrow l)$ na $(l \rightarrow k)$.
3. Pro proměnné X_k a X_l vypočítejte nové podmíněné pravděpodobnostní distribuce podle následujících vzorců:

$$\begin{aligned}
P(X_l | (X_i)_{i \in pa_{new}(l)}) &=^3 \\
&= \sum_{x \in \mathbf{X}_k} P(X_k = x | (X_i)_{i \in pa_{old}(k)}) \cdot P(X_l | (X_i)_{i \in pa_{old}(l) - \{k\}} \& X_k = x), \\
P(X_k | (X_i)_{i \in pa_{new}(k)}) &= \frac{P(X_k | (X_i)_{i \in pa_{old}(k)}) \cdot P(X_l | (X_i)_{i \in pa_{old}(l)})}{P(X_l | (X_i)_{i \in pa_{new}(l)})} = \\
&= \frac{P(X_k | (X_i)_{i \in pa_{old}(k)}) \cdot P(X_l | (X_i)_{i \in pa_{old}(l)})}{\sum_{x \in \mathbf{X}_k} P(X_k = x | (X_i)_{i \in pa_{old}(k)}) \cdot P(X_l | (X_i)_{i \in pa_{old}(l) - \{k\}} \& X_k = x)}.
\end{aligned}$$

Ilustrujme si tento postup (především pak použití pravidla otáčení hrany) na jednoduchém příkladu, který bude pravděpodobně názornější, než vysvětlení, že se zde vlastně nejedná o víc než o použití Bayesova vzorce.

Uvažujme bayesovskou síť z příkladu na str. 72 a předpokládejme, že naším cílem je spočítat podmíněnou distribuci $P(X_3 | X_1 = x_1, X_2 = x_2, X_5 = x_5)$ pro nějaké zadané hodnoty x_1, x_2, x_5 .



Použitím pravidla pro vypuštění uzlu můžeme nejprve vypustit uzel 7 a potom také postupně 6 a 4. Takže po aplikaci tohoto pravidla dostaneme bayesovskou síť zadanou

³Zde pracujeme vlastně se dvěma grafy: s grafem původním a s grafem vzniklým při transformaci. Proto používáme symboly pa_{old} a pa_{new} , abychom rozlišili, ve kterém grafu příslušnou množinu rodičů uvažujeme. Týká se to samozřejmě pouze uzlů k a l , neboť pro všechny ostatní uzly se množina rodičů nemění.

⁴Symbol \mathbf{X}_i zde označujeme množinu hodnot veličiny X_i .

grafem z předcházejícího obrázku (a) a příslušnými podmíněnými pravděpodobnostními distribucemi $P(X_1)$, $P(X_2)$, $P(X_3|X_1)$ a $P(X_5|X_2, X_3)$. Chceme-li počítat podmíněnou pravděpodobnost veličiny X_3 potřebujeme ještě otočit orientaci hrany ($3 \rightarrow 5$) na ($5 \rightarrow 3$). Po zdědění rodičů a otočení příslušné hrany tak dostaneme graf (b).

Pravděpodobnostní distribuce $P(X_1)$ a $P(X_2)$ se nemění, nicméně musíme vypočítat podle uvedených vzorců distribuce

$$P(X_5|X_1, X_2)$$

a

$$P(X_3|X_1, X_2, X_5).$$

Podle výrazu

$$P(X_5|X_1, X_2) = \sum_{x \in \mathbf{X}_3} P(X_3 = x|X_1) \cdot P(X_5|X_2, X_3 = x)$$

dostáváme

$$P(X_5 = 0|X_1 = 0, X_2 = 0) = \frac{3}{4}0 + \frac{1}{4}1 = \frac{1}{4}$$

$$P(X_5 = 1|X_1 = 0, X_2 = 0) = \frac{3}{4}1 + \frac{1}{4}0 = \frac{3}{4}$$

$$P(X_5 = 0|X_1 = 0, X_2 = 1) = \frac{3}{4}1 + \frac{1}{4}0 = \frac{3}{4}$$

$$P(X_5 = 1|X_1 = 0, X_2 = 1) = \frac{3}{4}0 + \frac{1}{4}1 = \frac{1}{4}$$

$$P(X_5 = 0|X_1 = 1, X_2 = 0) = \frac{1}{4}0 + \frac{3}{4}1 = \frac{3}{4}$$

$$P(X_5 = 1|X_1 = 1, X_2 = 0) = \frac{1}{4}1 + \frac{3}{4}0 = \frac{1}{4}$$

$$P(X_5 = 0|X_1 = 1, X_2 = 1) = \frac{1}{4}1 + \frac{3}{4}0 = \frac{1}{4}$$

$$P(X_5 = 1|X_1 = 1, X_2 = 1) = \frac{1}{4}0 + \frac{3}{4}1 = \frac{3}{4}.$$

Podobně, výraz

$$P(X_3|X_1, X_2, X_5) = \frac{P(X_3|X_1)P(X_5|X_2, X_3)}{\sum_{x \in \mathbf{X}_3} P(X_3 = x|X_1) \cdot P(X_5|X_2, X_3 = x)}$$

dává potřebné hodnoty distribuce $P(X_3|X_1, X_2, X_5)$, které si v následujícím spočítáme pouze pro $X_3 = 1$.

$$\begin{aligned}
P(X_3 = 1 | X_1 = 0, X_2 = 0, X_5 = 0) &= \frac{\frac{1}{4}1}{\frac{3}{4}0 + \frac{1}{4}1} = 1 \\
P(X_3 = 1 | X_1 = 0, X_2 = 0, X_5 = 1) &= \frac{\frac{1}{4}0}{\frac{3}{4}1 + \frac{1}{4}0} = 0 \\
P(X_3 = 1 | X_1 = 0, X_2 = 1, X_5 = 0) &= \frac{\frac{1}{4}0}{\frac{3}{4}1 + \frac{1}{4}0} = 0 \\
P(X_3 = 1 | X_1 = 0, X_2 = 1, X_5 = 1) &= \frac{\frac{1}{4}1}{\frac{3}{4}0 + \frac{1}{4}1} = 1 \\
P(X_3 = 1 | X_1 = 1, X_2 = 0, X_5 = 0) &= \frac{\frac{3}{4}1}{\frac{1}{4}0 + \frac{3}{4}1} = 1 \\
P(X_3 = 1 | X_1 = 1, X_2 = 0, X_5 = 1) &= \frac{\frac{3}{4}0}{\frac{1}{4}1 + \frac{3}{4}0} = 0 \\
P(X_3 = 1 | X_1 = 1, X_2 = 1, X_5 = 0) &= \frac{\frac{3}{4}0}{\frac{1}{4}1 + \frac{3}{4}0} = 0 \\
P(X_3 = 1 | X_1 = 1, X_2 = 1, X_5 = 1) &= \frac{\frac{3}{4}1}{\frac{1}{4}0 + \frac{3}{4}1} = 1.
\end{aligned}$$

6.3 Transformace na rozložitelný model

Jiný způsob práce s Bayesovskými sítěmi, který se později stal jakýmsi standardem, zavedli Steffen Lauritzen a David Spiegelhalter. Ten je založen na myšlence, že provádění výpočtů v Bayesovských sítích je algoritmicky složité, a proto je vhodné takovouto síť ještě před započítáním výpočtů transformovat do tvaru umožňujícího efektivnější zpracování. Tvarem, který tito autoři zvolili je takzvaný rozložitelný model, který vymezuje třídu pravděpodobnostních distribucí se zvláště výhodnými vlastnostmi. K jeho definici existuje několik ekvivalentních přístupů, z nichž zde zvolíme ten, který vychází z myšlenky *grafických modelů*.

Uvažujme tentokrát neorientovaný graf $G = (V, E)$. *Klikou* grafu G rozumíme každou maximální úplnou podmnožinu jeho uzlů⁵. Nechť $\{C_1, \dots, C_K\}$ je systém klik uvažovaného grafu G . Říkáme, že pravděpodobnostní distribuce *faktorizuje* vzhledem ke grafu G , jestliže existují reálné funkce ψ_1, \dots, ψ_K takové, že

$$P((X_i)_{i \in V}) = \prod_{k=1}^K \psi((X_i)_{i \in C_k}).$$

⁵ Jinými slovy, klika je podmnožina uzlů, v níž je každá dvojice spojena hranou. Požadavek maximality znamená, že přidáme-li k uvažované podmnožině ještě nějaký uzel, pak již tato rozšířená podmnožina nebude úplná a bude v ní tedy existovat dvojice uzlů, mezi kterými nebude hrana. Jestliže graf obsahuje izolované uzly, pak každý takovýto uzel tvoří samostatnou kliku.

Ne každá pravděpodobnostní distribuce faktorizující vzhledem k nějakému grafu je rozložitelným modelem. Tím jsou pouze distribuce, které faktorizují vzhledem k triangulovaným (rozložitelným) grafům. Neorientovaný graf $G = (V, E)$ nazveme *triangulovaným* jestliže každý jeho cyklus délky větší než 3 obsahuje (alespoň jednu) tětivu (t.j. hranu spojující dva uzly uvažovaného cyklu, které nejsou v cyklu bezprostředně zasebou).

V následujícím výkladu budeme potřebovat jednu důležitou větu z teorie grafů, kterou si zde uvedeme bez důkazu.

Tvrzení 5. Neorientovaný graf je triangulovaný právě tehdy, jestliže existuje uspořádání (očíslování) systému jeho klik (C_1, \dots, C_K) tak, že platí tzv. *running intersection property*:

$$\forall i = 2, \dots, K \exists 1 \leq j < i \quad (C_i \cap (\bigcup_{k=1}^{i-1} C_k) \subseteq C_j).$$

Transformace Bayesovské sítě do tvaru odpovídajícího rozložitelnému modelu probíhá v následujících 2 krocích.

Moralizace. V prvním kroku spojíme všechny dvojice uzlů, které mají společného přímého následníka (tedy “rodiče” stejného “dítěte”), neorientovanou hranou a všechny původně orientované hrany nahradíme hranami neorientovanými. Tím zkontruujeme neorientovaný graf, ve kterém pro každé $i \in V$ množina uzlů $pa(i) \cup \{i\}$ ⁶ tvoří úplný podgraf. To znamená, že pro každou takovou množinu uzlů existuje klika “moralizovaného” grafu, do které uvažovaná množina patří. Vzhledem k tomu, že na každou podmíněnou distribuci

$$P(X_i | (X_j)_{j \in pa(i)})$$

můžeme pohlížet jako na funkci proměnných $(X_j)_{j \in pa(i) \cup \{i\}}$, je očividné, že distribuce

$$P((X_i)_{i \in V}) = \prod_{i \in V} P((X_i) | (X_j)_{j \in pa(i)})$$

faktorizuje vzhledem k takto definovanému moralizovanému grafu.

⁶“Rodiče” $pa(i)$ uzlu i rozumíme samozřejmě vzhledem k původnímu orientovanému grafu definujícímu bayesovskou síť.

Triangularizace. V následujícím kroku doplníme (je-li třeba) do moralizovaného grafu některé hrany tak, aby výsledný graf byl triangulovaný. Tvzení, že jestliže distribuce faktorizuje vzhledem k nějakému grafu G , pak faktorizuje také vzhledem ke každému grafu, který je rozšířením grafu G , je triviální a ponecháváme jeho prověření čtenáři.

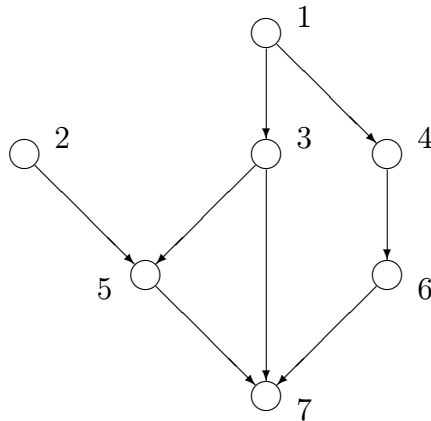
V tomto okamžiku jsme našli rozložitelný model pravděpodobnostní distribuce realizované původní bayesovskou sítí. Pomocí příslušného triangulovaného grafu umíme například vypočítat určité marginální distribuce bez zdoluhavého sečítání. Nicméně, pro efektivní výpočet jakékoliv podmíněné pravděpodobnosti neskončíme v této fázi, ale budeme ještě konstruovat příslušný *strom spojení* (junction tree).

Konstrukce stromu spojení. Nejprve zkonstruujeme základní strom spojení, jehož uzly jsou tvořeny klikami nalezeného triangulovaného grafu. K tomu, abychom věděli, které uzly máme propojit hranou použijeme předcházející tvrzení. Nalezneme uspořádání odpovídající “running intersection property” a pro všechna $i = 2, \dots, K$ uzel i spojíme hranou s příslušným uzlem j , pro který $(C_i \cap (\bigcup_{k=1}^{i-1} C_k)) \subseteq C_j$. Vzhledem k tomu, že každý uzel je tímto algoritmem propojen k předcházejícím uzlům právě jednou hranou, dostáváme souvislý graf obsahující přesně o jednu hranu méně, než je počet uzlů, a tedy graf je skutečně stromem.⁷

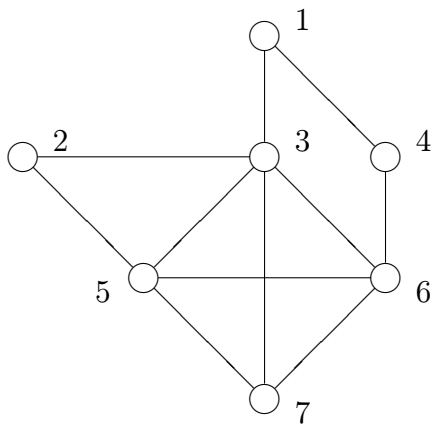
Takto vytvořený základní strom doplníme ještě další množinou tzv. *spojovacích uzlů* (uzly odpovídající klikám původního grafu budeme nazývat uzly *základními*). Doplnění provedeme tak, že každou hranu základního stromu spojení přerušíme a do prostředka umístíme uzel, kterému bude odpovídat množina uzlů, která je průnikem množin (klik) s novým uzlem sousedících.

Příklad. Ilustrujme si uvedený postup na příkladu ze strany 72. Tam jsme uvažovali bayesovskou síť se sedmi binárními náhodnými veličinami (X_1, \dots, X_7) a následujícím grafem.

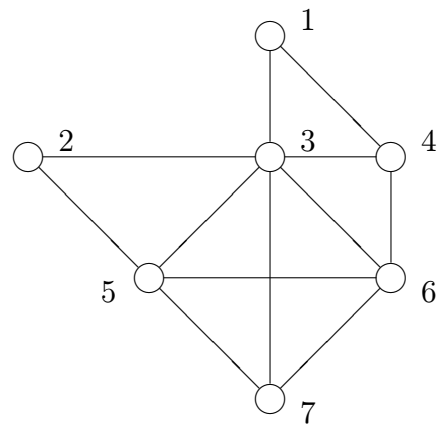
⁷Vzhledem k tomu, že ke každému triangulovanému grafu existuje několik uspořádání jeho klik splňujících “running intersection property”, není tímto postupem strom spojení určen jednoznačně. Všechny dále popsané postupy lze provádět pomocí kteréhokoliv z takto sestavených stromů spojení.



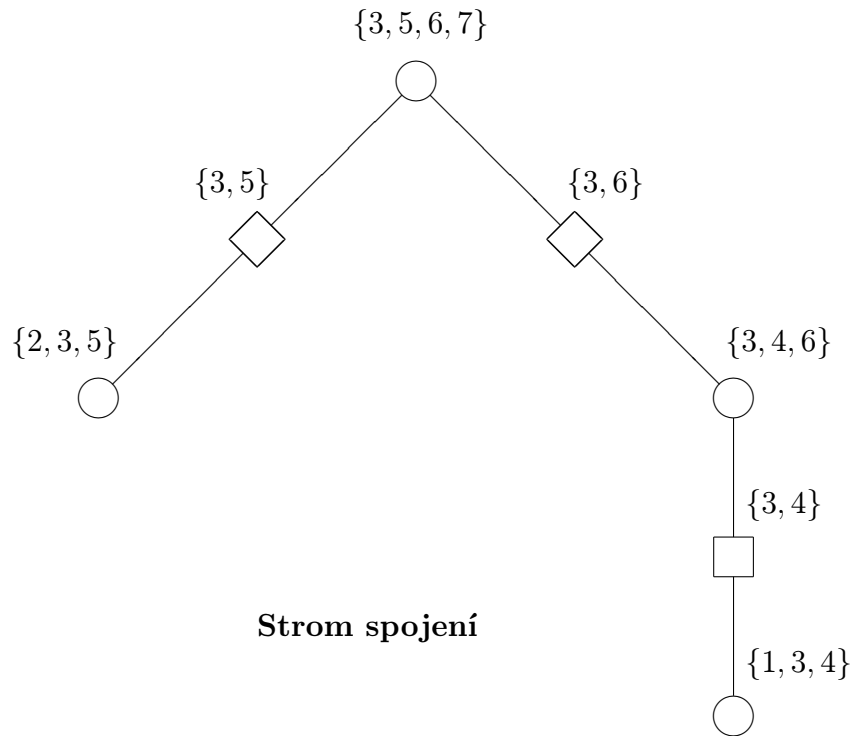
Ten se moralizací změní na neorientovaný graf, ve kterém jsou všechny uzly, které v původním grafu měly společné “děti”, spojeny hranami. Vzhledem k tomu, že v horní části moralizovaného grafu vznikl cyklus 1, 3, 6, 4 délky 4, musíme přidat do tohoto cyklu alespoň jednu “tětivu”, abychom dostali triangulovaný graf. K přidání přicházejí v úvahu hrany (1, 6) nebo (3, 4). Přidáním druhé z nich obdržíme již triangulovaný graf, který obsahuje čtyři kliky: $\{1, 3, 4\}$, $\{3, 4, 6\}$, $\{2, 3, 5\}$ a $\{3, 5, 6, 7\}$.



Moralizovaný graf



Triangularizovaný graf



Chceme-li nyní pro tento triangulovaný graf nalézt strom spojení, potřebujeme nejprve uvedené jeho čtyři kliky uspořádat tak, aby toto upořádání splňovalo “running intersection property”. Například výše uvedené pořadí $(\{1, 3, 4\}, \{3, 4, 6\}, \{2, 3, 5\}, \{3, 5, 6, 7\})$ tuto vlastnost nespĺňuje. Vezmeme-li uvedené kliky v pořadí zcela opačném, t.j. $(\{3, 5, 6, 7\}, \{2, 3, 5\}, \{3, 4, 6\}, \{1, 3, 4\})$, pak je již uvedená vlastnost splněna. Aplikací postupu pro konstrukci stromu spojení dostaneme jednoduchý strom uvedený výše.

6.4 Výpočty ve stromech spojení

Cílem tohoto odstavce je ukázat na příkladu, že s pomocí stromu spojení lze vypočítat libovolnou podmíněnou pravděpodobnost efektivním způsobem. I když to není nezbytné, využijeme zde jedno důležité tvrzení týkající se rozložitelných modelů, které si zde uvedeme opět bez důkazu. Nicméně důkaz tohoto tvrzení není příliš těžký a proto si jej může hloubavý čtenář provést sám.

Tvrzení 6. Každá pravděpodobnostní distribuce P faktorizující vzhledem k triangulovanému grafu $G = (V, E)$ může být vyjádřena ve tvaru následujícího součinu

$$P^{C_1} \frac{P^{C_2}}{P^{C_2 \cap C_1}} \frac{P^{C_3}}{P^{C_3 \cap (C_1 \cup C_2)}} \cdots \frac{P^{C_K}}{P^{C_K \cap (C_1 \cup \dots \cup C_{K-1})}},$$

jestliže systém klik C_1, \dots, C_K grafu G je uspořádán tak, aby splňoval “running intersection property”.

Toto tvrzení využijeme pro zvýšení názornosti následujících výpočtů. Uvažujme libovolnou pravděpodobnostní distribuci faktorizující vzhledem k triangulovanému grafu G s klikami C_1, \dots, C_K . Podle definice to znamená, že existují reálné funkce $\psi_1((X_i)_{i \in C_1}), \dots, \psi_K((X_i)_{i \in C_K})$ takové, že jejich součin definuje distribuci P . Vzhledem k tomu, že každá z podmíněných pravděpodobností

$$\frac{P^{C_k}}{P^{C_k \cap (C_1 \cup \dots \cup C_{k-1})}}$$

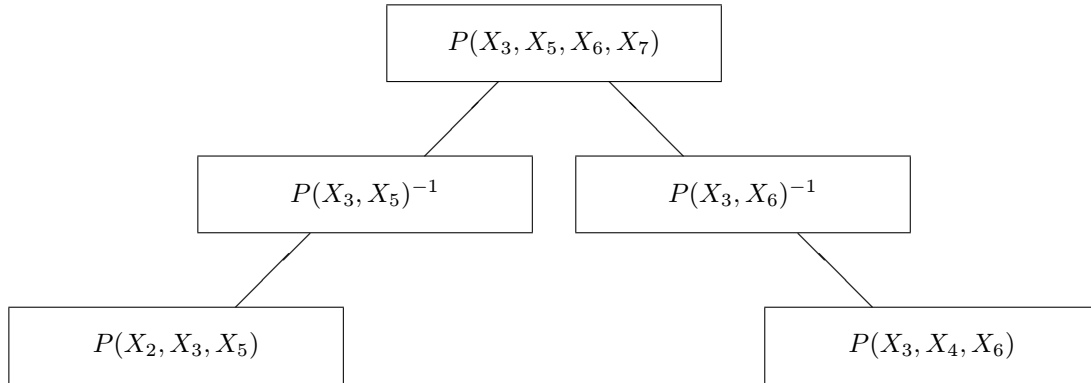
je vlastně také funkcí proměnných $(X_i)_{i \in C_k}$ vidíme, že pro rozložitelné modely, t.j. distribuce faktorizující vzhledem k triangulovaným grafům, můžeme volit funkce ψ_k ve speciálním tvaru podmíněných distribucí. Toho využijeme v následující ukázce, jak se počítají podmíněné pravděpodobnosti ve stromech spojení pomocí takzvaného *posílání zpráv*.

Princip těchto výpočtů spočívá ve skutečnosti, že každému uzlu stromu spojení (t.j. jak uzlům základním tak i uzlům spojovacím) jsou přiřazeny reálné funkce těch veličin, které jsou uzlu přiřazeny. Přitom jsou tyto funkce postupně přepočítávány tak, že součin všech těchto funkcí v každém časovém okamžiku se stále rovná pravděpodobnostní distribuci P .

Předpokládejme, že před započítáním výpočtu jsou základním uzlům spojovacího stromu přiřazeny příslušné marginální distribuce a uzlům spojovacím jsou přiřazeny odpovídající marginální distribuce umocněné na -1 . Vzhledem k tomu, že při konstrukci stromu jsme využívali skutečnost, že posloupnost klik splňuje “running intersection property”, je součin všech těchto funkcí přesně roven výrazu

$$P^{C_1} \frac{P^{C_2}}{P^{C_2 \cap C_1}} \frac{P^{C_3}}{P^{C_3 \cap (C_1 \cup C_2)}} \cdots \frac{P^{C_K}}{P^{C_K \cap (C_1 \cup \dots \cup C_{K-1})}}.$$

Omezíme-li se pro jednoduchost v následujících výpočtech pouze na část spojovacího stromu odpovídajícího klikám $\{3, 5, 6, 7\}$, $\{2, 3, 5\}$ a $\{3, 4, 6\}$, dostáváme přiřazení

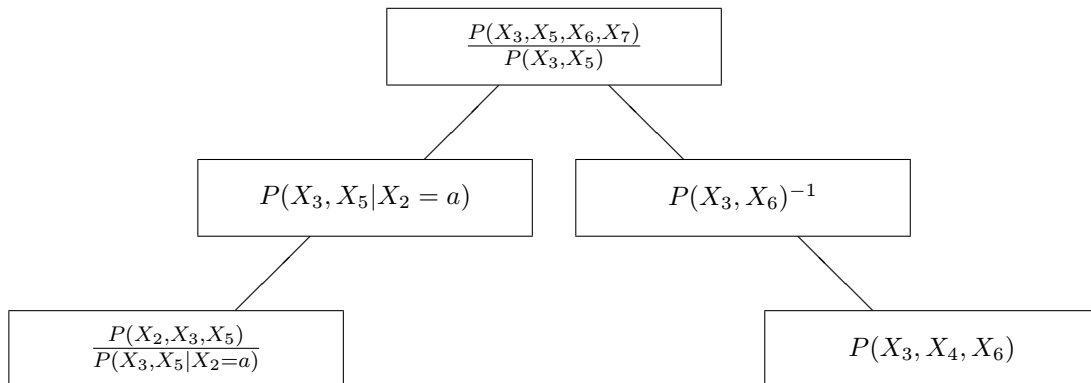


kterému odpovídá skutečně pravděpodobnostní distribuce

$$P(X_2, X_3, X_4, X_5, X_6, X_7) = \frac{P(X_3, X_5, X_6, X_7)P(X_2, X_3, X_5)P(X_3, X_4, X_6)}{P(X_3, X_5)P(X_3, X_6)}.$$

Předpokládejme nyní, že naším úkolem je spočítat podmíněnou pravděpodobnostní distribuci $P(X_7|X_2 = a, X_4 = b)$, pro nějaké konkrétní hodnoty a, b .

Základnímu uzlu $\{2, 3, 5\}$ je přiřazena pravděpodobnostní distribuce $P(X_2, X_3, X_5)$. Z ní snadno vypočteme podmíněnou distribuci $P(X_3, X_5|X_2 = a)$. Toto je funkce proměnných X_3 a X_5 a můžeme ji tedy přiřadit spojovacímu uzlu $\{3, 5\}$, který je sousedem uzlu $\{2, 3, 5\}$. Jelikož však je tomuto uzlu přiřazena funkce $P(X_3, X_5)^{-1}$, znásobíme nejprve touto funkcí distribuci přiřazenou uzlu $\{3, 5, 6, 7\}$ tak, aby součin všech funkcí zůstal zachován. Tím dostáváme nové přiřazení funkcí uzlům stromu spojení, nicméně jejich součin je zachován.



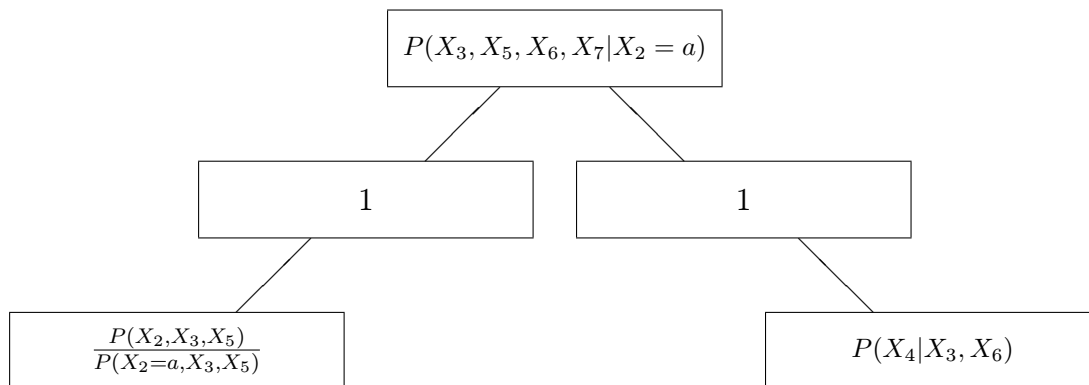
V následujícím kroku prostě vynásobíme funkce přiřazené uzlům $\{3, 5\}$ a $\{3, 5, 6, 7\}$ a výsledek, který je funkcí proměnných X_3, X_5, X_6, X_7 , přiřadíme uzlu $\{3, 5, 6, 7\}$. Vzhledem k tomu, že pravděpodobnostní distribuce P odpovídá původní bayesovské síti, víme, že veličiny (X_2) a (X_6, X_7) jsou podmíněně nezávislé při daných (X_3, X_5) . Proto

$$P(X_6, X_7 | X_2, X_3, X_5) = P(X_6, X_7 | X_3, X_5)$$

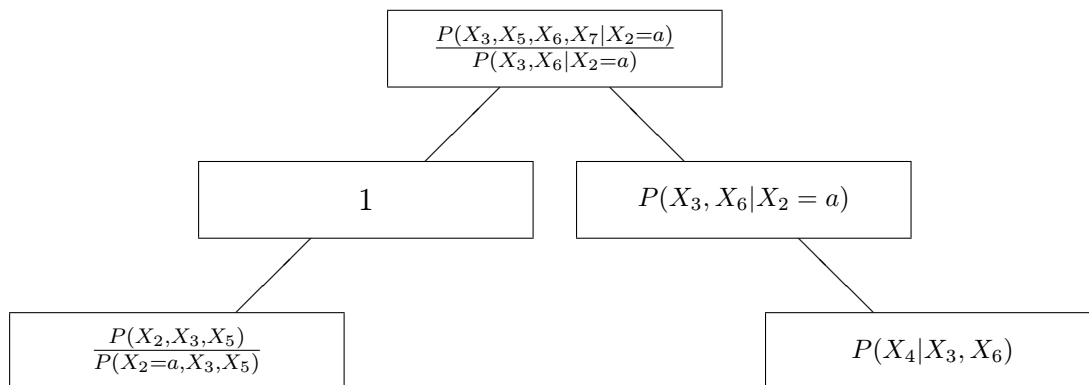
a tedy i

$$P(X_6, X_7, X_3, X_5 | X_2) = P(X_6, X_7 | X_3, X_5) P(X_3, X_5 | X_2).$$

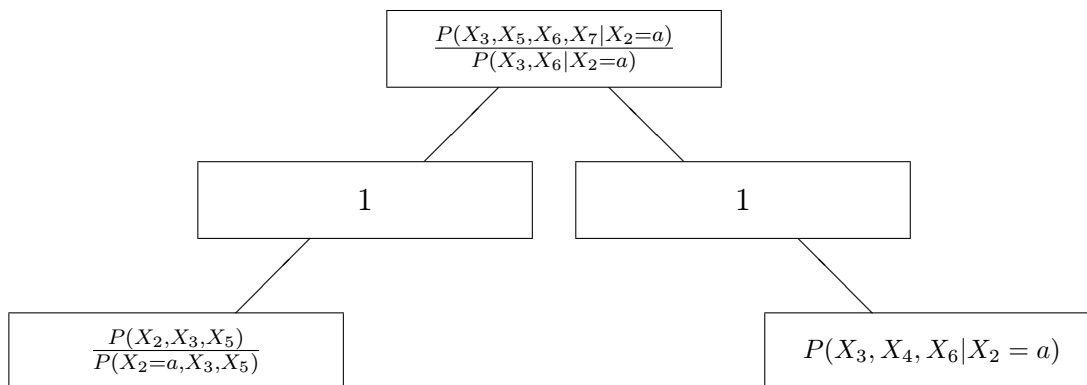
Pro ušetření času můžeme ještě v tomto kroku vynásobit funkce přiřazené uzlům $\{3, 6\}$ a $\{3, 4, 6\}$ a výsledek, který je funkcí proměnných X_3, X_4, X_6 , přiřadit uzlu $\{3, 5, 6, 7\}$.



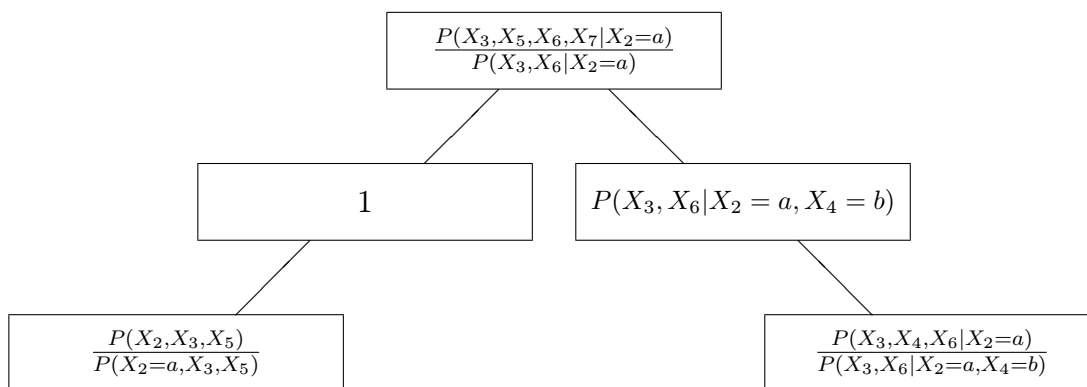
Nyní pouze z $P(X_3, X_5, X_6, X_7 | X_2 = a)$ vypočítáme $P(X_3, X_6 | X_2 = a)$ a přiřadíme tuto funkci spojovacímu uzlu $\{3, 6\}$.



Vynásobením $P(X_3, X_6|X_2 = a)$ a $P(X_4|X_3, X_6)$ dostáváme $P(X_3, X_4, X_6|X_2 = a)$, neboť opět vzhledem ke speciálnímu tvaru námi uvažované pravděpodobnostní distribuce P jsou veličiny (X_5, X_7) a (X_4) podmíněně nezávislé při zadaných veličinách (X_3, X_6) . Výslednou podmíněnou distribuci $P(X_3, X_4, X_6|X_2 = a)$, která je funkcí veličin (X_3, X_4, X_6) přiřazujeme základnímu uzlu $\{3, 4, 6\}$.



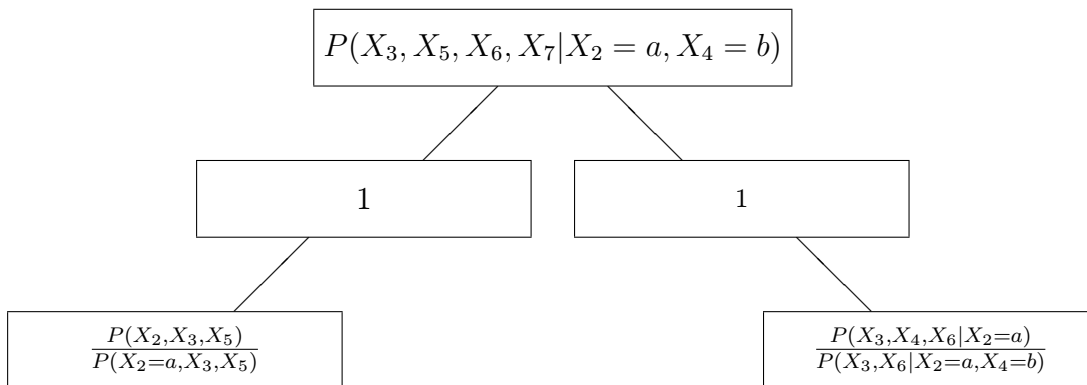
Z distribuce $P(X_3, X_4, X_6|X_2 = a)$, která je funkcí veličin (X_3, X_4, X_6) , vypočítáme podmíněnou distribuci $P(X_3, X_6|X_2 = a, X_4 = b)$, kterou můžeme přiřadit spojovacímu uzlu $\{3, 6\}$, neboť je funkcí pouze příslušných veličin.



K výpočtu požadované podmíněné pravděpodobnostní distribuce $P(X_7|X_2 = a, X_4 = b)$ již zbývá poslední krok. Opět si musíme uvědomit, že uvažovaná pravděpodobnostní distribuce je definovaná původní bayesovskou sítí a proto jsou veličiny (X_5, X_7) a (X_4) podmíněně nezávislé při daných (X_3, X_6) . Proto

$$P(X_3, X_5, X_6, X_7|X_2 = a, X_4 = b) =$$

$$= P(X_5, X_7 | X_3, X_6, X_2 = a) P(X_3, X_6 | X_2 = a, X_4 = b).$$



Požadovanou distribuci $P(X_7 | X_2 = a, X_4 = b)$ již snadno vypočítáme z $P(X_3, X_5, X_6, X_7 | X_2 = a, X_4 = b)$ sečítáním (marginalizací) přes zbývající tři veličiny.

Shrňme závěrem princip výpočtů, který jsme si zde ilustrovali. Metodu její autoři nazvali “posíláním zpráv”, protože se veškeré výpočty provádějí lokálně v jednom uzlu a výsledek je pak předáván pouze do uzlů sousedních. Přitom platí zásada, že pro výpočet jakékoliv podmíněné pravděpodobnosti stačí vždy “zprávu” vyslat z jednoho uzlu spojovacího stromu a počkat na odezvu od všech ostatních uzlů. To znamená, že výpočet proběhne v každém uzlu maximálně dvakrát. Z této vlastnosti lze i odvodit algoritmickou složitost výpočtů, která je lineární funkcí prostoru pořebného k uložení všech funkcí přiřazených uzlům stromu spojení.

6.5 Cvičení

1. Dokažte, že v acyklickém orientovaném grafu $G = (V, \vec{E})$ lze vždy uzly očíslovat

$$no : V \longrightarrow \{1, \dots, |V|\}$$

tak, že

$$i \in pa(j) \implies no(i) < no(j).$$

Návod: Tvrzení dokazujte indukcí přes počet uzlů grafu tak, že ukážete, že v každém acyklickém grafu existuje alespoň jeden uzel, který nemá žádného rodiče.

2. Využitím stejné myšlenky jako v předchozím cvičení dokažte, že pro každý uzel acyklického orientovaného grafu, který má alespoň jednoho potomka, existuje mezi jeho potomky alespoň jeden, do kterého nevede z uvažovaného uzlu jiná cesta než přímá hrana.
3. Dokažte, že vypuštění uzlu podle uvedeného pravidla skutečně vede k bayesovské síti definující marginál distribuce reprezentované původní bayesovskou sítí.
4. Dokažte, že otočení hrany podle uvedeného pravidla skutečně vede k bayesovské síti definujícímu stejnou distribuci jako původní bayesovská síť.
5. Užitím popsaného postupu spočtete $P(X_1|X_7)$ pro bayesovskou síť zadanou v příkladu na str. 72.
6. Navrhněte bayesovskou síť popisující situaci při házení hrací kostkou, kterou jsme uvažovali v příkladu na str. 62.
7. Dokažte, že jestliže existuje uspořádání (očíslování) (C_1, \dots, C_K) systému klik nějakého grafu pro které platí “running intersection property”

$$\forall i = 2, \dots, K \exists 1 \leq j < i \quad (C_i \cap (\bigcup_{k=1}^{i-1} C_k) \subseteq C_j),$$

potom je tento graf triangulovaný.

6.6 Literatura

1. Hájek, P., Havránek, T., Jiroušek, R.: Uncertain Information Processing in Expert Systems. *CRC Press, Inc., Boca Raton, 1992.*
2. Jensen, F.V.: Introduction to Bayesian Networks. Hugin Expert A/S, Aalborg, 1993.
3. Lauritzen, S.L., Spiegelhalter, D.J.: Local computations with probabilities on graphical structures and their application to expert systems (with discussion). *Journal of the Royal Statistical Society, Series B*, 50, pp 157-224 (1988).
4. Shachter, R. D.: Evaluating influence diagrams. *Operations Res.*, 34, pp 871-882 (1986)
5. Shachter, R. D.: Intelligent probabilistic inference. In: L.N. Kanal, J.F. Lemmer (eds.): *Uncertainty in Artificial Intelligence*. North-Holland, Amsterdam, 1986
6. Shachter, R. D.: Probabilistic inference and influence diagrams. *Operations Res.*, 36, pp 589-604 (1988)

Příloha A

Základní pojmy konečné teorie pravděpodobnosti

Tato příloha obsahuje definice a základní vlastnosti všech pojmů z teorie pravděpodobnosti, které jsou v textu použity. Vzhledem k tomu, že se čtenář se všemi těmito pojmy seznámil i v jiných předmětech (jako například ve statistice), nejsou do tohoto textu zařazeny žádné příklady.

A.1 Pravděpodobnostní rozložení

Nechť \mathbf{X} je konečná množina. *Pravděpodobnostní rozložení* na \mathbf{X} (často budeme též používat pojem *pravděpodobnostní distribuce*, abychom naznačili vztah k náhodným veličinám nabývajícím hodnot z \mathbf{X}) je zobrazení z potenční množiny $\mathcal{P}(\mathbf{X})$ ¹ množiny \mathbf{X} do uzavřeného intervalu $\langle 0, 1 \rangle$:

$$P : \mathcal{P}(\mathbf{X}) \longrightarrow \langle 0, 1 \rangle,$$

takové, že

1. $P(\mathbf{X}) = 1$ a
2. $A, B \subseteq \mathbf{X}, A \cap B = \emptyset \implies P(A \cup B) = P(A) + P(B)$.

Poznámka. Vlastnost číslo 2 je nazývána *aditivitou* pravděpodobnosti.

¹ $\mathcal{P}(\mathbf{X})$ je tedy množina všech podmnožin množiny \mathbf{X} .

Tvrzení. Pro libovolné pravděpodobnostní rozložení \mathbf{X} platí:

1. $P(\emptyset) = 0$,
2. $A \subseteq \mathbf{X}, \implies P(A) = 1 - P(\bar{A})$,²
3. $A, B \subseteq \mathbf{X}, A \subset B \implies P(A) \leq P(B)$.

Podmíněnou pravděpodobností jevu A při jevu B budeme nazývat číslo $P(A|B)$ pro které platí

$$P(A|B)P(B) = P(A \cap B).$$

Poznámka. Je-li B jev *nemožný*, t.j. je-li $P(B) = 0$, vyhovuje této definici jakékoliv číslo. Abychom se v tomto textu vyhnuli z toho vyplývajícím nepříjemnostem, budeme pro $P(B) = 0$ vždy uvažovat $P(A|B) = P(A)$.

Tvrzení.

1. Pro libovolný fixní jev B , pro který je $P(B) > 0$, je $P(A|B)$ pravděpodobnostním rozložením na \mathbf{X} .
2. Je-li $P(A) > 0$ a $P(B) > 0$, potom

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

Poznámka. Vzorec uvedený pod bodem 2 je nazýván *Bayesovým vzorcem*.

Jevy $A, B \subseteq \mathbf{X}$ jsou při pravděpodobnostním rozložení P *nezávislé*, jestliže

$$P(A \cap B) = P(A)P(B).$$

Symbolicky budeme tuto situaci zaznamenávat

$$A \perp_P B.$$

Obdobně, pro libovolné jevy $A_1, A_2, \dots, A_n \subseteq \mathbf{X}$ budeme symbolem

$$\perp_P (A_1, A_2, \dots, A_n)$$

² \bar{A} označuje doplněk množiny A .

označovat situaci, kdy jsou všechny tyto jevy A_1, A_2, \dots, A_n při pravděpodobnostním rozložení P *nezávislé*, což nastává, když

$$\forall I \subseteq \{1, 2, \dots, n\} \quad P\left(\bigcap_{i \in I} A_i\right) = \prod_{i \in I} P(A_i).$$

Poznámka. Stojí za povšimnutí, že pro to, aby jevy $A_1, A_2, \dots, A_n \subseteq \mathbf{X}$ byly nezávislé nestačí, aby

$$P\left(\bigcap_{i=1}^n A_i\right) = \prod_{i=1}^n P(A_i).$$

Kdyby totiž byla pro definici nezávislosti zvolena tato zjednodušená podmínka, potom by libovolná skupina jevů obsahující alespoň jeden nemožný jev byla nezávislá.

Tvrzení.

1. Pro libovolné pravděpodobnostní rozložení P a libovolný jev $A \subseteq \mathbf{X}$

$$A \perp_P \emptyset, \quad A \perp_P \mathbf{X}.$$

2. Je-li $P(B) = 0$, pak $A \perp_P B$ pro libovolné $A \subseteq \mathbf{X}$. Jinak

$$A \perp_P B \iff P(A|B) = P(A).$$

- 3.

$$A \perp_P B \iff A \perp_P \overline{B}.$$

Nyní uvažujme tři obecné jevy $A, B, C \subseteq \mathbf{X}$ a nějaké pravděpodobnostní rozložení P na \mathbf{X} . Budeme říkat, že jevy A a B jsou *podmíněně nezávislé při daném C* a značit

$$A \perp_P B|C,$$

jestliže

$$P(A \cap B|C) = P(A|C)P(B|C).^3$$

Povšimněme si, že při naší úmluvě týkající se podmiňování nemožným jevem tato definice při $C = \emptyset$ degeneruje na nepodmíněnou nezávislost. Tohoto faktu budeme často využívat.

Samozřejmě, že lze tento pojem přirozeně rozšířit na podmíněnou nezávislost libovolného počtu jevů. Nicméně, v dalším textu toto zobecnění nebudeme potřebovat, proto

³Pro jistotu připomeňme, že výrazem $P(A \cap B|C \cup D)$ rozumíme $P((A \cap B)|(C \cup D))$.

ponecháváme na rozhodnutí čtenáře, zda si jej provede sám.

Tvrzení. Pro libovolné pravděpodobnostní rozložení P a libovolné jevy $A, B, C \subseteq \mathbf{X}$ je následujících pět výrazů ekvivalentních:

1. $A \perp_P B|C$,
2. $P(A|B \cap C) = P(A|C)$,
3. $P(B|A \cap C) = P(B|C)$,
4. $P(A \cap B \cap C) = P(A \cap C)P(B|C)$,
5. $P(A \cap B \cap C) = P(B \cap C)P(A|C)$.

A.2 Náhodné veličiny

Nyní se začneme zabývat speciální situací, kdy množina \mathbf{X} bude kartézským součinem množin hodnot, kterých mohou nabývat náhodné veličiny X_1, X_2, \dots, X_n . Potom pravděpodobnostní rozložení P definované na

$$\mathbf{X}_1 \times \mathbf{X}_2 \times \dots \times \mathbf{X}_n$$

budeme též nazývat *pravděpodobnostní distribucí náhodných veličin* X_1, X_2, \dots, X_n .

Uvažujme nejprve pro jednoduchost pouze dvě náhodné veličiny X_1, X_2 a nějakou jejich *sdrúženou* pravděpodobnostní distribucí $P(X_1, X_2)$. Je zřejmé, že z této distribuce můžeme (díky aditivitě pravděpodobnostního rozložení) spočítat pravděpodobnost

$$P(X_1 = a) = P(\{(x_1, x_2) \in \mathbf{X}_1 \times \mathbf{X}_2 : x_1 = a\}) = \sum_{x_2 \in \mathbf{X}_2} P(X_1 = a, X_2 = x_2)$$

odpovídající *jednorozměrné marginální distribuci* náhodné veličiny X_1 . Zobecněním této úvahy dostaneme, že libovolná n -rozměrná pravděpodobnostní distribuce definuje jednoznačně všechny své méněrozměrné marginální distribuce. Nicméně poznamenejme, že opačné tvrzení neplatí. V obecném případě jakýkoliv systém marginálních distribucí neurčuje sdrúženou distribuci jednoznačně.

Víme-li, co je distribuce marginální k dvourozměrné distribuci $P(X_1, X_2)$, můžeme si definovat *podmíněnou pravděpodobnostní distribuci* veličiny X_1 při dané veličině X_2 jako funkci $P(X_1|X_2)$ (opět na kartézském součinu $\mathbf{X}_1 \times \mathbf{X}_2$, respektive na jeho potenční množině), pro kterou platí

$$P(X_1|X_2)P(X_2) = P(X_1, X_2).$$

Obdobně, pro dvě disjunktní podmnožiny indexů $I_1, I_2 \subset \{1, 2, \dots, n\}$ definujeme podmíněnou pravděpodobnostní distribuci $P((X_i)_{i \in I_1} | (X_i)_{i \in I_2})$ výrazem

$$P((X_i)_{i \in I_1} | (X_i)_{i \in I_2})P((X_i)_{i \in I_2}) = P((X_i)_{i \in I_1 \cup I_2}).$$

Je-li $I = \emptyset$, potom definujeme $P(X_i)_{i \in I} = 1$.

Poznámka. Z uvedené dodatečné poznámky vyplývá, že pro $I_2 = \emptyset$, degeneruje podmíněná distribuce $P((X_i)_{i \in I_1} | (X_i)_{i \in I_2})$ opět na nepodmíněnou distribuci $P((X_i)_{i \in I_1})$.

Tvrzení.

1. Pro libovolnou hodnotu $a \in \mathbf{X}_2$ je $P(X_1 | X_2 = a)$ pravděpodobnostní distribucí veličiny X_1 .
2. Je-li $P(X_1, X_2) > 0$, potom

$$P(X_1 | X_2) = \frac{P(X_2 | X_1)P(X_1)}{\sum_{x \in \mathbf{X}_1} P(X_2 | X_1 = x)P(X_1 = x)}.$$

Poznámka. Vzorec uvedený pod bodem 2 je nazýván *Bayesovým vzorcem pro náhodné veličiny*. Jestliže si čtenář není jistý jak tomuto vzorci rozumět, poznamenejme, že se jedná o rovnost dvou funkcí, neboli, že pro všechny dvojice $(x_1, x_2) \in \mathbf{X}_1 \times \mathbf{X}_2$ musí platit

$$P(X_1 = x_1 | X_2 = x_2) = \frac{P(X_2 = x_2 | X_1 = x_1)P(X_1 = x_1)}{\sum_{x \in \mathbf{X}_1} P(X_2 = x_2 | X_1 = x)P(X_1 = x)}.$$

Analogicky k pojům, které jsme si zavedli při studiu pravděpodobností jednotlivých jevů (podmnožin \mathbf{X}), budeme si nyní definovat nezávislost a podmíněnou nezávislost náhodných veličin.

Uvažujme opět nejprve dvourozměrnou pravděpodobnostní distribuci $P(X_1, X_2)$. Veličiny X_1 a X_2 jsou při distribuci P *nezávislé* jestliže

$$P(X_1, X_2) = P(X_1)P(X_2).$$

Jak lze očekávat, budeme fakt, že veličiny X_1 a X_2 jsou při distribuci P nezávislé označovat symbolem $X_1 \perp_P X_2$.

Pro vícerozměrné náhodné veličiny se nabízí velice obecná alternativa tohoto pojmu. My si zde zavedeme pouze její dvě speciálnější verze. Budeme definovat situaci, kdy

všechny veličiny jsou nezávislé a kdy se veličiny rozdělí na dvě vzájemně nezávislé skupiny.

Uvažujme tedy obecně n -rozměrnou náhodnou veličinu $P(X_1, \dots, X_n)$. Budeme říkat, že veličiny X_1, \dots, X_n jsou při distribuci P *vzájemně nezávislé* jestliže

$$P(X_1, \dots, X_n) = \prod_{i=1}^n P(X_i).$$

Tuto nezávislost budeme označovat symbolem

$$\perp_P ((X_i)_{i=1, \dots, n}).$$

Poznámka. Všimněme si rozdílu v této definici nezávislosti více veličin oproti situaci, když jsme si definovali nezávislost více jevů.

Disjunktní skupiny náhodných veličin $(X_i)_{i \in I_1}$ a $(X_i)_{i \in I_2}$ (pro $I_1, I_2 \subset \{1, 2, \dots, n\}$, $I_1 \cap I_2 = \emptyset$) jsou při distribuci P *nezávislé* (značíme $(X_i)_{i \in I_1} \perp_P (X_i)_{i \in I_2}$), jestliže

$$P((X_i)_{i \in I_1 \cup I_2}) = P((X_i)_{i \in I_1})P((X_i)_{i \in I_2}).$$

Jsou-li $I_1, I_2, I_3 \subset \{1, 2, \dots, n\}$ po dvou disjunktní, ($I_1 \neq \emptyset \neq I_2$), potom říkáme, že $(X_i)_{i \in I_1}$ a $(X_i)_{i \in I_2}$ jsou *podmíněně nezávislé* při $(X_i)_{i \in I_3}$, jestliže

$$P((X_i)_{i \in I_1 \cup I_2} | (X_i)_{i \in I_3}) = P((X_i)_{i \in I_1} | (X_i)_{i \in I_3})P((X_i)_{i \in I_2} | (X_i)_{i \in I_3}).$$

Tuto situaci budeme symbolicky označovat

$$(X_i)_{i \in I_1} \perp_P (X_i)_{i \in I_2} | (X_i)_{i \in I_3}.$$

Poznámka. Obdobně jako na Bayesův vzorec musíme se i na výše uvedené rovnosti dívat jako na rovnosti funkcí.

Tvrzení. Pro libovolné pravděpodobnostní rozložení P a libovolné neprázdné disjunktní podmnožiny indexů $I_1, I_2 \subset \{1, 2, \dots, n\}$ jsou následující tři výrazy ekvivalentní:

1. $(X_i)_{i \in I_1} \perp_P (X_i)_{i \in I_2}$,
2. $P((X_i)_{i \in I_1} | (X_i)_{i \in I_2}) = P((X_i)_{i \in I_1})$,

$$3. P((X_i)_{i \in I_2} | (X_i)_{i \in I_1}) = P((X_i)_{i \in I_2}).$$

Tvrzení. Jestliže pro nějaké pravděpodobnostní rozložení $P(X_1, \dots, X_n)$ je

$$\perp_P ((X_i)_{i=1, \dots, n}),$$

pak jsou nezávislé veličiny i jeho marginálu $P((X_i)_{i \in I})$ pro libovolné $I \subseteq \{1, \dots, n\}$, t.j.

$$\perp_P ((X_i)_{i \in I}).$$

Tvrzení. Pro libovolné pravděpodobnostní rozložení P a libovolné disjunktní podmnožiny indexů $I_1, I_2, I_3 \subset \{1, 2, \dots, n\}$ ($I_1 \neq \emptyset \neq I_2$) je následujících pět výrazů ekvivalentních:

1. $(X_i)_{i \in I_1} \perp_P (X_i)_{i \in I_2} | (X_i)_{i \in I_3}$,
2. $P((X_i)_{i \in I_1} | (X_i)_{i \in I_2 \cup I_3}) = P((X_i)_{i \in I_1} | (X_i)_{i \in I_3})$,
3. $P((X_i)_{i \in I_2} | (X_i)_{i \in I_1 \cup I_3}) = P((X_i)_{i \in I_2} | (X_i)_{i \in I_3})$,
4. $P((X_i)_{i \in I_1 \cup I_2 \cup I_3}) = P((X_i)_{i \in I_1 \cup I_3}) P((X_i)_{i \in I_2} | (X_i)_{i \in I_3})$,
5. $P((X_i)_{i \in I_1 \cup I_2 \cup I_3}) = P((X_i)_{i \in I_2 \cup I_3}) P((X_i)_{i \in I_1} | (X_i)_{i \in I_3})$.

A.3 Pojmy teorie informace

Budiž P pravděpodobnostní rozložení definované na konečné množině \mathbf{X} . *Shannonovskou entropií* $H(P)$ tohoto rozložení nazveme číslo

$$H(P) = - \sum_{x \in \mathbf{X}} P(x) \log P(x).$$

Obdobně *Shannonovskou entropií pravděpodobnostní distribuce* $P(X_1, \dots, X_n)$ definuje číslo

$$H(P(X_1, \dots, X_n)) = - \sum_{x_1 \in \mathbf{X}_1} \dots \sum_{x_n \in \mathbf{X}_n} P(x_1, \dots, x_n) \log P(x_1, \dots, x_n).$$

Tvrzení. Pro libovolné pravděpodobnostní rozložení P na \mathbf{X} platí

1. $0 \leq H(P) \leq \log |\mathbf{X}|$.

2. $H(P) = 0$ právě když existuje $x \in \mathbf{X}$, pro které $P(x) = 1$. (Všechny ostatní body množiny \mathbf{X} jsou tedy jevy nemožnými.)
3. $H(P) = \log |\mathbf{X}|$ právě když

$$P(x) = \frac{1}{|\mathbf{X}|}$$

pro všechny body $x \in \mathbf{X}$.

Vzájemná informace obsažená ve sdružené dvourozměrné pravděpodobnostní distribuci $P(X_1, X_2)$ je definována

$$I_P(X_1; X_2) = \sum_{x_1 \in \mathbf{X}_1} \sum_{x_2 \in \mathbf{X}_2} P(x_1, x_2) \log \frac{P(x_1, x_2)}{P(x_1)P(x_2)}.$$

Máme-li obecně n -rozměrnou pravděpodobnostní distribuci $P(X_1, \dots, X_n)$, potom zobecněním tohoto pojmu je *vzájemná informace mezi dvěma skupinami veličin* $(X_i)_{i \in I_1}$ a $(X_i)_{i \in I_2}$ ($I_1 \cap I_2 = \emptyset$, $I_1 \cup I_2 = \{1, \dots, n\}$) definovaná výrazem

$$\begin{aligned} I_P((X_i)_{i \in I_1}; (X_i)_{i \in I_2}) &= \\ &= \sum_{x_1 \in \mathbf{X}_1} \dots \sum_{x_n \in \mathbf{X}_n} P(x_1, \dots, x_n) \log \frac{P(x_1, \dots, x_n)}{P((x_i)_{i \in I_1})P((x_i)_{i \in I_2})}. \end{aligned}$$

Tvrzení. Pro libovolnou n -rozměrnou pravděpodobnostní distribuci $P(X_1, \dots, X_n)$ a jakýkoliv netriviální rozklad $\{I_1, I_2\}$ indexové množiny $\{1, \dots, n\}$ platí

1. $0 \leq I_P((X_i)_{i \in I_1}; (X_i)_{i \in I_2}) \leq \min_{k=1,2} [H(P((X_i)_{i \in I_k}))]$.
2. $I_P((X_i)_{i \in I_1}; (X_i)_{i \in I_2}) = 0$ právě když $((X_i)_{i \in I_1}) \perp_P ((X_i)_{i \in I_2})$.
3. $I_P((X_i)_{i \in I_1}; (X_i)_{i \in I_2}) = H(P((X_i)_{i \in I_1}))$ právě když $((X_i)_{i \in I_1})$ je při distribuci P funkčně závislé na $((X_i)_{i \in I_2})$, t.j., když pro každou kombinaci hodnot

$$((x_i)_{i \in I_2}) \in \times_{i \in I_2} \mathbf{X}_i$$

existuje právě jedna kombinace hodnot

$$((x_i)_{i \in I_1}) \in \times_{i \in I_1} \mathbf{X}_i$$

taková, že

$$P((X_i = x_i)_{i \in I_1} | (X_i = x_i)_{i \in I_2}) = 1.$$

Poslední pojem, kterým se budeme zabývat je *informační obsah* sdružené pravděpodobnostní distribuce $P(X_1, \dots, X_n)$. Ten je definován výrazem

$$I_P(X_1, \dots, X_n) = \sum_{x_1 \in \mathbf{X}_1} \dots \sum_{x_n \in \mathbf{X}_n} P(x_1, \dots, x_n) \log \frac{P(x_1, \dots, x_n)}{P(x_1) \dots P(x_n)}.$$

Tvrzení. Pro libovolnou n -rozměrnou pravděpodobnostní distribuci $P(X_1, \dots, X_n)$ platí

1. $0 \leq I_P(X_1, \dots, X_n)$.
2. $I_P(X_1, \dots, X_n) = 0$ právě když $\perp_P ((X_i)_{i=1, \dots, n})$.
3. Pro libovolný netriviální rozklad $\{I_1, I_2\}$ (t.j. $I_1 \neq \emptyset \neq I_2$) indexové množiny $\{1, \dots, n\}$ platí

$$I_P(X_1, \dots, X_n) = I_P((X_i)_{i \in I_1}) + I_P((X_i)_{i \in I_2}) + I_P((X_i)_{i \in I_1}; (X_i)_{i \in I_2})$$

a tedy

$$I_P(X_1, \dots, X_n) = I_P((X_i)_{i \in I_1}) + I_P((X_i)_{i \in I_2})$$

právě když

$$((X_i)_{i \in I_1}) \perp_P ((X_i)_{i \in I_2}).$$

A.4 Cvičení

1. Dokažte všechna tvrzení uvedená v tomto dodatku.

A.5 Literatura

1. Hájek, P., Havránek, T., Jiroušek, R.: Uncertain Information Processing in Expert Systems. *CRC Press, Inc., Boca Raton, 1992.*

Rejstřík

- Croneckerovo delta, 48
- dominance rozhodovacích funkcí, 48
- heuristická funkce, 13
- heuristický algoritmus, 14
- modifikovaný algoritmus pro výpočet lineární rozhodovací funkce, 42
- pravděpodobnost chyby rozhodovací funkce $p_d(a_1 \rightarrow a_2)$, 48
- pravděpodobnost chyby rozhodovací funkce $p_d(a_2 \rightarrow a_1)$, 48
- pravděpodobnost dosažení uzlu, 28
- pravděpodobnostní míra, 91
- princip maximální entropie, 74
- rozhodovací funkce realizovaná rozhodovacím stromem - optimalita, 25
- rozhodovací strom - střední délka, 28
- rozhodovací tabulka, 21
- rozklad množiny - zjemnění, 31
- Shannonovská entropie, 53
- zobrazení totální, 7

- aditivita pravděpodobnosti, 91
- algoritmus A^* , 14
- algoritmus pro konstrukci rozhodovacího stromu z rozhodovací tabulky, 26
- algoritmus pro konstrukci rozhodovacího stromu využívající princip maximálního snižování entropie, 30
- algoritmus pro výpočet lineární rozhodovací funkce, 37
- algoritmus slepého prohledávání orientovaného grafu, 11
- alternativa, 21
- alternativa rozlišitelná v rozhodovací tabulce, 22
- antecedent pravidla, 57

- bayesovská síť, 67
- bayesovská síť - úpravy, 75
- Bayesův vzorec pro jevy, 92
- Bayesův vzorec pro náhodné veličiny, 95

- cena operátoru, 10
- cena řešení, 10
- cesta v grafu, 11

- detektor, 21
- délka řešení, 8

- faktORIZACE pravděpodobnostní distribuce, 79
- FIFO, 12

- graf neorientovaný, 79
- graf orientovaný, 11
- graf orientovaný acyklický, 57, 67
- graf rozložitelný, 80
- graf stavového prostoru, 11
- graf triangulovaný, 80

- heuristika, 13
- hrana orientovaná, 11

- IF-THEN pravidlo, 57
- influenční diagram, 67
- informační obsah sdružené distribuce, 99

- jev nemožný, 92
- junction tree, 81

- kauzální síť, 67
- klika grafu, 79
- kombinace odpovědí konzistentní se sloupcem rozhodovací tabulky, 22
- kořen stromu, 23

- kriteriální funkce, 31
- LIFO, 12
- list, 23
- marginální pravděpodobnostní distribuce, 94
- metoda korekcí, 60
- metoda postupných modifikací, 75
- metoda posílání zpráv, 84
- metoda větví a mezí, 31
- metoda větví a mezí - aplikace na konstrukci optimálního rozhodovacího stromu, 32
- moralizace bayesovské sítě, 80
- nejistota pravidla, 58
- nezávislost jevů, 92
- nezávislost jevů podmíněná, 93
- nezávislost veličin, 95
- náhodná veličina, 94
- náhodné veličiny podmíněně nezávislé, 96
- následník uzlu, 11
- odpověď, 19
- odvozovací pravidlo PROSPECTORovské, 59
- odvozovací pravidlo PROSPEKTORovské - pravděpodobnostní korektnost, 60
- operátor, 7
- operátor aplikovatelný na stav, 7
- optimální řešení, 31
- otočení hrany v bayesovské síti, 76
- otázka, 19
- otázka binární, dichotomická, 21
- očíslování uzlů bayesovské sítě, 68
- podmíněná nezávislost jevů, 93
- podmíněná pravděpodobnost, 92
- podmíněná pravděpodobnostní distribuce, 95
- podmíněně nezávislé náhodné veličiny, 96
- potenční množina, 91
- pravděpodobnostní distribuce marginální, 94
- pravděpodobnostní distribuce náhodných veličin, 94
- pravděpodobnostní distribuce podmíněná, 95
- pravděpodobnostní distribuce sdružená, 94
- pravděpodobnostní rozložení, 91
- pravidlo expertního systému, 57
- princip maximální entropie, 53
- princip maximální neurčitosti, 53
- princip maximálního snižování entropie, 30
- princip minimaxu, 53
- prohledávání slepé, 12
- prohledávání slepé, do hloubky, 12
- prohledávání slepé, do šířky, 12
- prohledávání řízené, 12, 13
- PROSPECTOR, 57
- pseudobayesovské skládání nejistot, 57
- pseudobayesovský přístup, 60
- předpoklad nezávislosti pro bayesovské sítě, 68
- ridič uzlu, 67
- riziko rozhodovací funkce, 52
- rozhodnutí, 21
- rozhodovací funkce, 21
- rozhodovací funkce konzistentní s rozhodovací tabulkou, 22
- rozhodovací funkce lineární, 35
- rozhodovací funkce optimální bayesovské, 52
- rozhodovací funkce optimální vzhledem ke kritériu maximální věrohodnosti, 52
- rozhodovací funkce optimální z hlediska celkového rizika, 52
- rozhodovací funkce přípustná, 49
- rozhodovací funkce realizovaná rozhodovacím stromem, 25

- rozhodovací strom, 23
- rozhodovací strom - použití, 24
- rozhodovací tabulka jednoznačná, 22
- rozhodování za neurčitosti, 53
- rozklad množiny, 31
- running intersection property, 80

- Schwartzova nerovnost, 39
- sružená pravděpodobnostní distribuce, 94
- Shannonovská entropie, 30, 74
- sloupec rozhodovací tabulky konzistentní kombinací odpovědí, 22
- sloupec rozhodovací tabulky konzistentní s odpovědí r , 22
- spojovací uzly, 81
- stav, 7
- stav cílový, 7
- stav počáteční, 7
- stavový prostor, 7
- strom, 23
- strom spojení, 81
- strom spojení - konstrukce, 81
- strom strom, 23
- stupeň nutnosti, 58
- stupeň postačitelnosti, 58
- střední chyba rozhodovací funkce, 52
- střední délka rozhodovacího stromu - dolní odhad, 30
- sukcedent pravidla, 57

- transformace bayesovské sítě, 75
- triangularizace, 81
- tvrzení o distribuci reprezentované bayesovskou sítí, 68
- tvrzení o distribucích faktorizujících vzhledem ke triangulovanému grafu, 84
- tvrzení o efektivitě algoritmu pro výpočet lineární rozhodovací funkce, 38
- tvrzení o optimalitě heuristického algoritmu A^* , 14
- tvrzení o přípustnosti rozhodovacích funkcí, 50
- tvrzení o rozložitelných distribucích, 84
- tvrzení o uspořádání klik v triangulovaném grafu, 80

- uzel neterminální, 23
- uzel terminální, 23
- uzel zdrojový, 67
- uzel základní, 81

- vypouštění uzlu z bayesovské sítě, 76
- vzájemná informace, 98
- výpočty ve stromech spojení, 83

- znalost deklarativní, 5
- znalost neuvědomnělá, 6
- znalost procedurální, 5
- znalosti kvalitativní, 68
- znalosti kvantitativní, 68
- znalosti strukturální, 68
- zobrazení aplikovatelné na stav, 7
- zobrazení parciální, 7
- zobrazení použitelné na stav, 7
- ztrátová funkce, 52

- šance, 58
- šance aposteriorní, 58
- šance apriorní, 59

- řešení, 7
- řešení cenově optimální, 10
- řešení optimální, 10

- úloha, 7
- úroveň uzlu, 23