EVROPSKÁ UNIE
MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
OP Vzdělávání pro konkurenceschopnost
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Vysoká škola báňská – Technická univerzita Ostrava

METODY ANALÝZY DAT

Učební text

Jana Šarmanová

Ostrava 2012

Recenze: prof. RNDr. Alena Lukasová, CSc.

Název: Metody analýzy dat
Autor: Jana Šarmanová
Vydání: první, 2012
Počet stran: 170

Studijní materiály pro studijní obor Informační a komunikační technologie FEI
Jazyková korektura: nebyla provedena.

Určeno pro projekt:
Operační program Vzděláváním pro konkurenceschopnost
Název: Personalizace výuky prostřednictvím e-learningu
Číslo: CZ.1.07/2.2.00/07.0339
Realizace: VŠB – Technická univerzita Ostrava
Projekt je spolufinancován z prostředků ESF a státního rozpočtu ČR

ISBN 978-80-248-2565-6

POKYNY KE STUDIU

Metody analýzy dat

Pro předmět Metody analýzy dat oboru Informační a komunikační technologie jste obdrželi studijní balík obsahující

- integrované skriptum pro distanční studium obsahující i pokyny ke studiu,
- doplňkové animace a programy vybraných částí kapitol,
- harmonogram průběhu semestru a rozvrh prezenční části.

Prerekvizity

Pro studium tohoto předmětu se předpokládá absolvování předmětu Informační systémy a datové sklady.

Cíl předmětu

Při řešení výzkumných i praktických úloh vzniká často problém s vyhodnocením a interpretací informací, které poskytují naměřená či evidovaná data z praxe. Metody analýzy dat, zvané též metody dolování znalostí z dat (Data Mining), seznamují studenty s řadou statistických a analytických metod pro řešení této třídy výzkumných i praktických problémů. Dále informují studenty o některých programových balících pro řešení těchto úloh. V posledních letech se tyto metody výrazně uplatňují při dolování znalostí z informací uložených v rozsáhlých databázích.

Po prostudování modulu by měl student být schopen samostatně analyzovat předložená data získaná dotazníkovými akcemi, účelovým sběrem dat pro výzkum některé oblasti nebo data získaná z existujících databází nebo datových skladů. Výsledkem jsou nejen základní statistické charakteristiky, ale i nové hypotézy, získané a platné v analyzovaných datech.

Pro koho je předmět určen

Modul je zařazen do magisterského studia oboru Informační a komunikační technologie na FEI, ale může jej studovat i zájemce z kteréhokoliv jiného oboru, pokud splňuje požadované prerekvizity.

Skriptum se dělí na části, kapitoly, které odpovídají logickému dělení studované látky, ale nejsou stejně obsáhlé. Předpokládaná doba ke studiu kapitoly se může výrazně lišit, proto jsou velké kapitoly děleny dále na číslované podkapitoly a těm odpovídá níže popsaná struktura.

Při studiu každé kapitoly doporučujeme následující postup:

Čas ke studiu: xx hodin

Na úvod kapitoly je uveden **čas** potřebný k prostudování látky. Čas je orientační a může vám sloužit jako hrubé vodítko pro rozvržení studia celého předmětu či kapitoly. Někomu se čas může zdát příliš dlouhý, někomu naopak. Jsou studenti, kteří se s touto problematikou ještě nikdy nesetkali a naopak takoví, kteří již v tomto oboru mají bohaté zkušenosti.

Cíl: Po prostudování tohoto odstavce budete umět

- popsat ...
- definovat ...
- vyřešit ...

Ihned potom jsou uvedeny cíle, kterých máte dosáhnout po prostudování této kapitoly – konkrétní dovednosti, znalosti.

Výklad

Následuje vlastní výklad studované látky, zavedení nových pojmů, jejich vysvětlení, vše doprovázeno obrázky, tabulkami, řešenými příklady, odkazy na animace.

Shrnutí pojmů 1.1.

Na závěr kapitoly jsou zopakovány hlavní pojmy, které si v ní máte osvojit. Pokud některému z nich ještě nerozumíte, vraťte se k nim ještě jednou.

Otázky 1.1.

Pro ověření, že jste dobře a úplně látku kapitoly zvládli, máte k dispozici několik teoretických otázek.

Úlohy k řešení 1.1.

Protože většina teoretických pojmů tohoto předmětu má bezprostřední význam a využití v databázové praxi, jsou Vám nakonec předkládány i praktické úlohy k řešení. V nich je hlavní význam předmětu a schopnost aplikovat čerstvě nabyté znalosti při řešení reálných situací hlavním cílem předmětu.

Úspěšné a příjemné studium s touto učebnicí Vám přeje autorka výukového materiálu

Jana Šarmanová

OBSAH

1. DOLOVÁNÍ ZNALOSTÍ Z DAT

Čas ke studiu: 2 hodiny

Cíl Po prostudování této kapitoly budete umět

- definovat a vysvětlit pojem dolování znalostí z dat
- popsat a zdůvodnit etapy procesu získávání znalostí a etapy dolování
- popsat využití výsledků dolování dat z různých oblastí reality

Výklad

1.1. Data, informace, znalost

❑ Realita jako prostor znaků

Svět, realita je pojem příliš obecný a široký. Reálnou skutečnost poznáváme po malých částech, často se navzájem překrývajících podle toho, proč nás právě tato část světa zajímá. Předměty našeho dílčího zájmu nazýváme **objekty**. Mohou to být lidé, zvířata, věci, jevy, procesy, vztahy či závislosti mezi nimi. Objekty nejčastěji poznáváme a popisujeme pomocí jejich vlastností - **atributů**, znaků, příznaků, údajů, v matematických disciplínách je pak nazýváme proměnnými. *(Chceme-li popsat člověka, o kterém budeme mluvit, řekneme například: ten malý blonďák v červené bundě ze 2. patra).* Výběr atributů pro dobrý popis objektů je obvykle problémem, který budeme diskutovat níže. Ideální by bylo, kdybychom v případě potřeby dalších údajů o pozorovaných objektech je mohli jednoduše získat. To však není obvykle reálné.

Ve smyslu výše uvedených úvah formulujme následující předpoklad:

Každá množina reálných objektů a jevů má své zákonitosti, své zařazení do hierarchie světa, svou klasifikaci na podtypy, své vztahy k okolí. Také podmnožiny atributů mohou mít mezi sebou důležité vztahy – asociace: korelace, příčiny a následky, skryté faktory apod.

Poznávání všech těchto skutečností je naším cílem.

❑ Poznávání reality

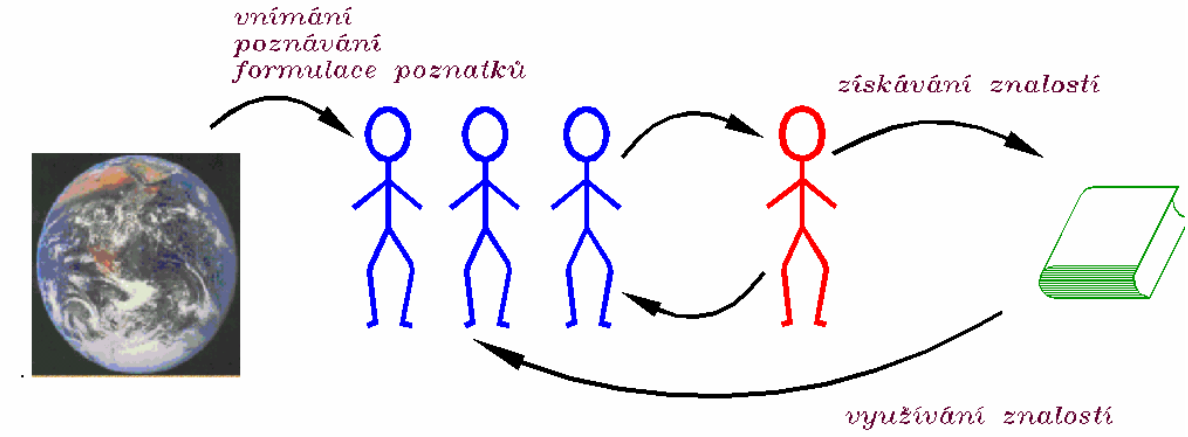
Dávno předtím, než dolování v datech dostalo své jméno, lidé získávali své znalosti pozorováním okolního světa a zobecňováním jednotlivých poznatků: rozdělováním pozorovaných objektů do podobných skupin, formulováním pravidel o tom, že má-li objekt vlastnosti A, má většinou i vlastnosti X; objeví-li se nový objekt s vlastnostmi B, patří nejspíš ke skupině C atd.

Pomineme dobu, kdy hlavním paměťovým médiem byla lidská paměť a získané znalosti byly předávány ústně (některé jsou dodnes, od přísloví a pranostik až po "nesahej na žehličku, pálí"). Jakmile začnou hloubavější z nás zkoumat některé jevy systematicky, často začínají shromažďováním

údajů a zkoumáním toho, jaká fakta o údajích platí. Ověřují, jestli se z faktů dají formulovat obecně platná pravidla, nebo dokonce dokázat některé (přírodní, společenské, …) zákonitosti.

Ve výzkumné praxi při zkoumání objektů zatím příliš málo známých či příliš složitých, u nichž dosud neumíme popsat jejich vlastnosti a chování, často také vycházíme z jejich pozorování. Sbíráme nebo měříme o nich údaje, které pozorovat můžeme, provádíme experimenty s různě nastavenými podmínkami apod. Tak o nich nashromáždíme množství údajů a z nich se pokoušíme vydedukovat jejich další, obecnější nebo skrytější vlastnosti.

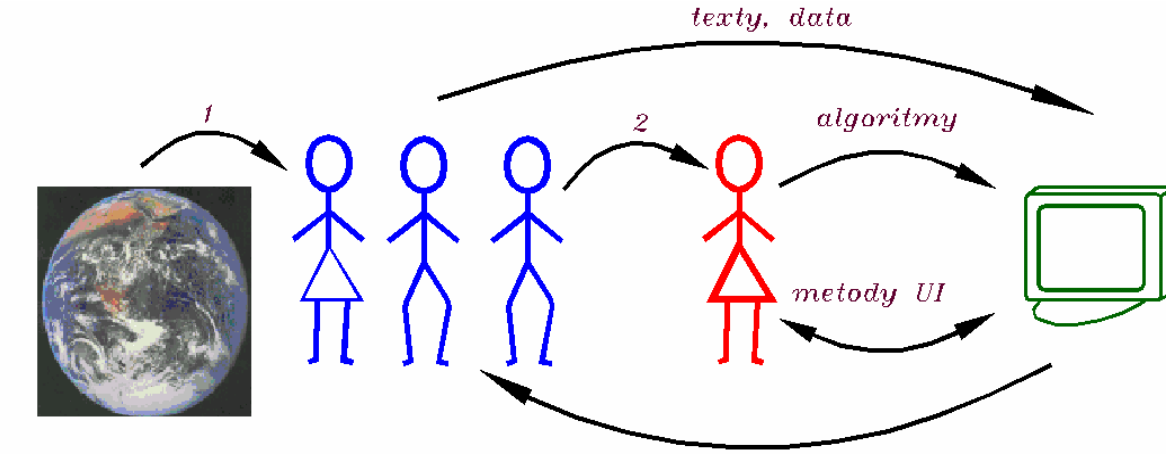
Tento postup prováděli lidé odpradávna. Zprvu jim za paměťové médium sloužila jen vlastní hlava, později papír, ještě později paměť počítačů. Pro vyhodnocování takových údajů a odvozování nových sloužil zprvu jen selský rozum, později hlavně matematické disciplíny - především matematická logika a statistika a konečně mnohé metody explorační analýzy dat, umělé inteligence, případně neuronových sítí. Metody analýzy dat vznikaly již dávno v době předpočítačové pro výzkum v nejrůznějších oborech (psychologie, biologie, technika atd.), doba počítačů jim však teprve dala velké možnosti využití.



Obrázek 1.1. Cyklus klasického poznávání světa

Sbírání údajů a jejich vyhodnocení používá nejen vědecký výzkum, ale i jednodušší úlohy, například sociologické průzkumy pro účely politické, reklamní, marketingové, dopravní, psychologické atd.

Víme, že k ověřování formulovaných hypotetických pravidel již dlouho slouží matematická statistika. Není ale obecně známo, že i pro etapu zobecňování jednotlivých faktů – pro etapu formulování hypotéz nad nimi, existuje již mnoho desetiletí řada metod.



Obrázek 1.2. Automatizované poznávání reality

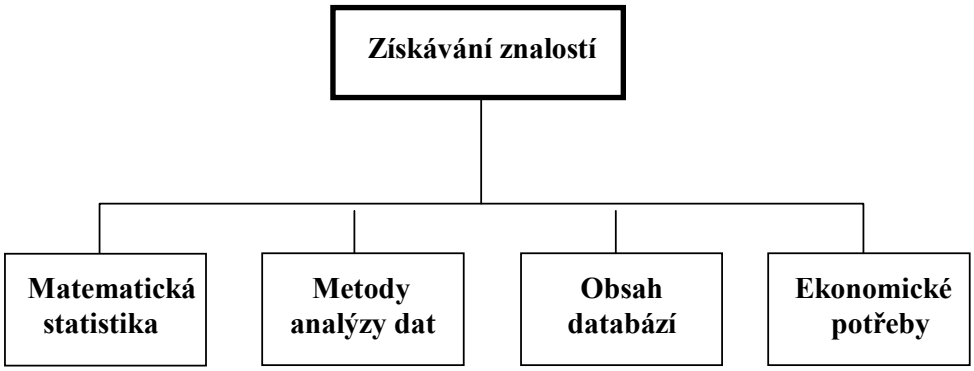
Nezávisle na tomto zkoumání člověk ve shodě s potřebou udržovat pořádek v evidencích o kdečem – o skupinách lidí, věcí, jevů apod., zakládal databáze údajů. Když údaje zestárly, zrušil je nebo archivoval a ukládal do evidence údaje nové. Největší rozsah údajů pravděpodobně postupně nashromáždily firmy o svých výrobách, obchodech, … Dobrému fungování firem pomáhají znalosti z ekonomie. Ovšem nejen firmy mají v databázích mnoho evidovaných dat, také nemocnice, školy, zemědělci, technici, sociologové, výzkumníci nejrůznějších oborů.

❑ Získávání znalostí jako multioborová disciplina

Teprve počátkem 90. let 20. století přišel nápad využít především údajů z počítačových databází, byť určených původně jen k evidenčním účelům, také jako zdroj automatizovaného získávání znalostí. Jsou k tomu využívány staré známé metody i metody nově vznikající.

Hlavním impulsem pro rozvoj nového oboru byl zájem firem zpracovávat svá data za účelem získání lepších informací o fungování firmy a umět tak lépe a rychleji reagovat na potřeby trhu, být konkurenceschopnější. (*Zvažme příklad obchodní firmy, evidující některé osobní údaje o svých zákaznících a jejich nákupech – době nákupu, seznamu zboží, způsobu platby. Analýzou takových údajů získají například skupiny podobných „typických" zákazníků a mohou jim efektivněji na míru nabízet své reklamní či věrnostní akce, čímž ušetří náklady na masovou reklamu a v zákaznících posílí dojem, že jsou tou správnou firmou pro jejich potřeby).*

Na rozdíl od ostatních majitelů dat velké firmy mohly rozvoj metod a jimi získané informace zaplatit. Tím výrazně přispěly ke vzniku nového integrovaného oboru, jeho pojmenováním byl proces vzniku završen.



Obrázek 1.3. Zdrojové obory pro disciplínu získávání znalostí z dat

Již jsme si uvedli, že mnohé úlohy a metody zařazované k dolování dat vznikly před řadou desetiletí. Byly nazývány **metodami analýzy mnohorozměrných dat**. Konkrétně se o historii každé metody zmíníme u jednotlivých úloh. V současnosti se pro celou disciplínu, patřící do počítačových věd, používá několik názvů. Z angličtiny převzatý **Knowledge Discovery in Databases** (KDD) nebo u nás i v originále používaný **Data Mining** (DM), v překladu pak v několika provedeních jako **získávání znalostí z dat** (jakýchkoliv, nejen databázových) či **získávání znalostí z databází,** také **dolování dat** nebo některé podobné názvy. Pokud se metody DM používají k získání užitečných znalostí z firemních dat, schopných podpořit rozvoj firmy, používá se často názvu **Business Intelligence** (BI).

Níže si uvedeme, že vlastní metody získávání znalostí z dat jsou součástí většího procesu, zahrnujícího výběr a přípravu dat, vlastní dolování, vizualizaci a interpretaci výsledků.

Statistika a metody získávání znalostí z dat

Mějme tedy následující obecnou úlohu: předmětem našeho zájmu jsou reálné objekty či jevy, k jejich zkoumání máme k dispozici jejich množinu (zřídka úplnou, obvykle více nebo méně rozsáhlý vzorek objektů, často jen jejich náhodný výběr). Každý objekt je popsán vektorem svých atributů, které získáme zaznamenáním, pozorováním, měřením, experimentováním apod. Úkolem je zjistit o objektech, jejich množině a jejich atributech vše, co se na základě dat zjistit dá.

Pro metody analyzující data budeme dále používat zkratku DM (Data-miningové metody). Protože je možno zkoumat data z různých hledisek, existuje i celá řada typů metod analýzy dat. Dávají výsledky různých typů, společný mají zdroj informací. DM jako disciplína je někdy zařazována do mnohorozměrné statistiky, jindy mezi metody umělé inteligence. Obě zařazení mají svá opodstatnění, protože některé konkrétní metody patří více tam, jiné onam.

Někdy se v této souvislosti mluví o rozdělení metod patřících k matematické statistice (též někdy ke konfirmační analýze dat) nebo k explorační analýze dat (DM).

Konfirmační analýza

- zahrnuje metody klasické matematické statistiky,
- používá metody, odpovídající na analytikovy otázky typu "je pravda, že ... ", tj. testují zadané hypotézy nad daty,
- nahrazují neznámé matematické vztahy mezi veličinami zadaným typem funkce, testují míru takové závislosti apod.,
- analytik formuluje požadavek, statistika mu dá na tento požadavek odpověď; na nevyslovené otázky neodpoví,
- v procesu poznávání světa patří na začátek etapy formulování nových znalostí.

Explorační analýza (explore = prozkoumat, probádat)

- provádějí tzv. orientační studie nad daty,
- samy si automaticky generují otázky zadaného typu a ihned testují výsledky,
- dávají odpovědi s menší vahou, ale odpovídají i na otázky nevyslovené typu "co je v datech zajímavé, neprůměrné, ... ",
- objevují i nové poznatky,
- jako výsledek formulují hypotézy podporované zkoumanými daty a předkládají je k dalšímu bádání, ověřování, testování,
- patří spíš k metodám umělé inteligence, v procesu poznávání světa patří do etapy poznávání, zobecňování.

Informace získané z databází

Databází tedy budeme chápat nejen firemní databáze, případně nad nimi vytvořené datové sklady, shromažďující a integrující i data archivní a další, ale jakákoliv data uspořádaná do (relačních) tabulek, kde v řádcích tabulky jsou údaje o jednom z množiny sledovaných objektů, ve sloupcích jsou atributy těchto objektů. Pro jednoduchost budeme předpokládat jedinou, někdy dokonce nenormalizovanou tabulku.

Je dána tabulka **X** popisující množinu objektů $\mathbf{O} = \{O_1, O_2, \ldots, O_n\}$ zadaných pomocí atributů $\mathbf{A} = \{A_1, A_2, \ldots, A_m\}$, každý s doménou z $\mathbf{D} = \{D_1, \ldots D_m\}$.

Z těchto dat je možno získat užitečné informace prostředky klasických informačních systémů. Jsou to různé komentované výběry dat, agregované hodnoty – sumy, průměry atd., jejich řady – časové, prostorové atd., jejich hierarchie. Všechny vznikají na základě požadavků uživatele, jejich nalezení je součástí IS nebo existuje jiná možnost získat, bývají doplněny vhodným formátováním, grafickým zobrazením. I vyšší úroveň databází, datové sklady, nabízejí uživatelům předzpracovaná data.

Mimo to však mohou být v datech rozptýleny další, dosud nepoznané skutečnosti, z nichž některé mohou být užitečné člověku a doplnit jeho dosavadní znalost světa.

Příklad 1.1. *Je dána tabulka dat za posledních 10 let o studentech gymnázia se strukturou:*

Student (skol_rok, trida, rod_cis, jmeno, vek, vyska, vaha, znam_CJ, …, skok_vys, …)

Tabulka obsahuje redundance, například opakující se (rod_cis, jmeno).

Klasickými SQL dotazy získáme informace o jednotlivých studentech a jejich atributech, jako například jakou známku má Jiří Novák z matematiky, jaký je průměr známek z matematiky v jednotlivých třídách apod.

Metodami matematické statistiky dostaneme odpověď i na další otázky, jako například „existuje korelace mezi známkami z matematiky a hudební nauky?“ a mnohé další.

Z datového skladu nad touto tabulkou můžeme zobrazit například časový průběh průměrných známek z matematiky v jednotlivých ročnících nebo podle jednotlivých učitelů atd.

Metodami získávání znalostí z dat pak můžeme objevit i další, v datech rozptýlené a na první pohled neviditelné zákonitosti, jako například „má-li student známku = 1 z občanské nauky, pak skok_vys > 110 cm se spolehlivostí 70%”. Na takovou souvislost se pravděpodobně nikdo nebude ptát a přesto se v datech může objevit,

♦

❑ Typy úloh dolování dat

Zopakujme si, že realitu poznáváme prostřednictvím reálných objektů a jejich atributů. Odhalování a poznávání vztahů mezi nimi je naším cílem.

Úlohy DM můžeme rozdělit z několika hledisek:

- ❑ Podle období platnosti výsledků na úlohy
  - **deskriptivní,** popisující obecné vlastnosti zadaných dat (*například jejich rozdělení do skupin podobných objektů, platné vztahy mezi jejich atributy apod.*)
  - **prediktivní**, umožňující na základě analýzy zadaných dat předpovídat chování – vlastnosti dat budoucích (*například nalezením pravidel, kterými se řídila skupina zákazníků dosud předpovědět jejich chování v následujícím období a přizpůsobit tak nabídku).*
- ❑ Jinou klasifikací můžeme rozdělit metody DM na
  - **charakterizace dat**, popisující obecné vlastnosti analyzovaných množin objektů nebo atributů; *(například najdeme silný vztah příčina - následek mezi atributy dlouholetý kuřák a onemocnění jednou z nemocí rakovina plic nebo infarkt),*
  - **diskriminace dat**, vyhledávající pro některé podmnožiny objektů nebo atributů, v čem se liší od ostatních dat, v čem jsou výjimečné; budeme je též nazývat analýzou výjimek *(například hledáme, co způsobuje sice velmi řídký, ale nepříjemný výskyt kazů materiálu při výrobě).*

- Podle toho, který typ vztahů zkoumají, rozdělujeme úlohy DM na 3 nejčastěji používané typy úloh:
  - **asociační**, hledající vztahy mezi podmnožinami atributů,
  - **shlukovací**, hledající vztahy mezi objekty, rozdělení objektů do podobných podmnožin a případně hierarchii těchto podmnožin,
  - **klasifikační**, hledající pravidla, podle nichž se data rozdělují do definovaných tříd.

Na základě všech výše uvedených informací budeme rozumět následující, nejčastěji uváděné definici.

Definice 1.

Dolováním znalostí z dat nazýváme proces netriviálního získávání implicitní, dříve neznámé a potencionálně užitečné informace z dat.

Netriviálním procesem rozumíme to, že informaci nelze získat jednoduše, například vhodným SQL dotazem, ale je nutné použít vhodnou speciální metodu. *Dříve neznámá* informace je informace v datech skrytá, rozptýlená, neviditelná na první pohled. *Potenciálně užitečná* informace je ta, která má význam jako nová vědecká hypotéza, jako podklad pro manažerské rozhodnutí, jako upozornění na neobvyklou skutečnost apod.

Rozdíl mezi OLAP a dolováním znalostí

Někdy se zaměňuje nebo nedostatečně rozlišuje OLAP a metody dolování znalostí. Obě skupiny metod provádějící výpočty nad daty mají za úkol vyhledávat pro uživatele nové užitečné informace.

OLAP je soubor metod, které provádějí výpočty uživatelem požadované a prezentují výsledky ve srozumitelné a publikovatelné podobě (tabulky, grafy, slovní formulace apod.). Pracují téměř výhradně nad datovým skladem. Jejich podstatou jsou převážně agregované hodnoty různých hierarchických stupňů. Z nich se vytváří časové řady, řady podél jiných dimenzí nebo podél více dimenzí současně. Tytéž výsledky je možno prezentovat na různých hierarchických úrovních a různými metodami. Úkolem je maximálně zpřístupnit pochopení výsledků uživateli a maximálně zjednodušit ovládání programu.

K dolování znalostí patří mnoho metod, které vyhledávají v datech nové, dosud neznámé a také nedotazované znalosti různých typů. Pracují (podle implementace) nad různými formáty dat, do DS jsou zařazovány spíše jako nadstavba, produkující něco navíc proti metodám OLAP.

Příklad 1.2.

Mějme část datového skladu banky s daty za 10 let evidence o poskytování a splácení úvěrů. Nástroji OLAP se může uživatel – marketingový ředitel – dotazovat na

- *dlouhodobý časový vývoj poskytovaných úvěrů a splátek celkem, podle zákazníků apod.,*
- *rozložení zákazníků a jejich splátek podle regionů, podle velikosti firem apod.*

Pokud má k dispozici i příslušné metody dolování, může například pomocí nich zjistit, že

- *zákazníci s úvěrem nad 50 mil. a s ručením některé státní organizace nesplácejí,*
- *že nesplácejí ještě některé další specifické podmnožiny zákazníků*

♦

1.2. Využití dolování znalostí

❑ Potřeby uživatelů metod dolování

Nejčastěji uváděná definice dolování znalostí z dat jako "proces netriviálního získávání implicitní, dříve neznámé a potencionálně užitečné informace z dat" je často doplňována sloganem "za účelem získání obchodní výhody".

V našem širším pohledu na tento proces rozdělíme potřeby dolování z hlediska uživatelů na

- průzkum - marketing, bankovnictví, výroba, pojišťovnictví, … (získání obchodních výhod)
- výzkum – medicína, biologie, hutnictví, … (získání nových odborných znalostí, hypotéz)
- sociologický průzkum – veřejné mínění, sčítání lidu, lokální věcné problémy, … (získání politických výhod)

Každá z těchto skupin má jiné nároky a potřeby na data, proces jejich dolování, míru spolehlivosti výsledků, prezentaci apod. Můžeme si je představit následovně:

	Marketing	Výzkum	Sociologie
DATA			
Zdroj	**Databáze**	**Sběr + databáze**	**Sběr**
Rozsah	**Velký**	**Menší**	**Malý**
Přírůstky	**Časté**	**Řídké – žádné**	**Žádné**
Struktura	**Stálá**	**Různá**	**Různá**
ZPRACOVÁNÍ			
Předzpracování	**Automatické**	**Na míru**	**Žádné**
Analýza potřeb	**Jednorázová**	**Na míru**	**Standardní**
Rychlost	**Vysoká-on line**	**Menší**	**Menší**
Úplnost, kvalita	**Informativní**	**Vysoká**	**Informativní**
Výstupy	**Graf, tabulka, text**	**Pracovní**	**Graf, tabulka, text**
Výsledky	**Aktuální**	**Dlouhodobé**	**Aktuální**
UŽIVATEL			
Primární	**Manažer**	**Výzkumník**	**Sociolog**
Sekundární		**Odborná veřejnost**	**Veřejnost**

To vše znamená, že datové sklady firemní a pro byznys – inteligenci na jedné straně a datové sklady pro využití v ostatních oblastech se budují rozdílně. Přitom současná literatura se zabývá téměř výhradně první kategorií, i když příklady užití (obvykle pro lepší pochopení čtenářů) čerpá odjinud.

1.3. Životní cyklus procesu získávání znalostí z dat

❑ Etapy procesu získávání znalostí z dat

Dolováním znalostí nazýváme proces hledání konkrétních typů znalostí pomocí konkrétních metod. Ovšem to je jen částí rozsáhlejšího procesu, do něhož se spojují v integrovaný celek i další části.

Celý proces je nazýván procesem vyhledávání znalostí v databázích (KDD). Tvoří jakýsi životní cyklus komplexní analýzy dat, obsahující postupně etapy formulace řešeného problému a problémové analýzy, datové analýzy - výběru (nebo sběru) relevantních dat, jejich předzpracování - integrace do

jednotného formátu, transformace a odvozování dat, dále vlastní dolování nových hypotéz, interpretace výsledků a konečně využití a zhodnocení celého procesu.

Obdobně, jako u každé složité činnosti, jsou pro projekty dolování znalostí definována mnohá teoretická pravidla: etapy projektu, řešitelské týmy, metody, algoritmy, SW nástroje, způsoby prezentace a interpretace výsledků, doporučení dalších postupů. Prakticky všechny se shodují v následujících etapách a následujících typech spoluřešitelů:

1. **Formulace problému** znamená věcné zadání výzkumné úlohy. Specifikuje podstatu úseku reality, která bude zkoumána a účel, pro který bude zkoumána. Provádí ji odborník na problémovou oblast (dále jen **expert**, řešitel). Vychází z potřeby přeměnit data na užitečnou informaci a znalost.

2. **Věcná analýza** úlohy se dělí na **datovou** analýzu (výběr objektů a jejich atributů relevantních vzhledem k zadanému účelu zkoumání, způsob kódování atributů nečíselných ap.) a analýzu **problémovou** (formulace problémů, formalizovaných otázek nebo jejich typů). Provádí ji odborník znalý dat (dále jen expert nebo též **majitel dat**), vhodná je konzultace s analytikem - odborníkem na metody analýzy dat (dále jen **analytik**). Výsledkem je seznam atributů každého sledovaného objektu i s jejich obory hodnot, často ve formě formuláře, dotazníku, tabulky ap., do nichž se údaje budou zaznamenávat. Dalším výsledkem je návrh metod, které budou aplikovány.

3. **Sběr údajů** je etapa z formálního hlediska zřejmá. Prakticky může znamenat třeba jen jednoduché sesbírání údajů (např. sociologický výzkum na základě vyplněných dotazníků), nebo provádění náročných experimentů (např. technických, technologických, biologických), nebo dlouhodobé sbírání údajů z praxe (např. v medicíně, meteorologii), výběr relevantních údajů z databáze (nejčastěji dat komerčních, ale i všech jiných typů) atd. Existují techniky a kritéria pro plánování experimentu, metody pro sběr dat, výběr reprezentativních vzorků ap., které tuto etapu podporují. Nasbírané údaje se zaznamenají pro další automatizované zpracování do počítače.

4. **Hrubá filtrace** dat je etapa předcházející vlastním výpočtům nad daty a nutná pro správnost výsledků. Jde o vyhledání dat chybných, chybějících, irelevantních, redundandních apod. Chyby mohou vzniknout na několika místech sběru: přímo při experimentu, při záznamu o jeho výsledku, při přepisu do počítače. Chybějící nenaměřené údaje některé metody následujícího zpracování nepřipouštějí, proto je třeba řešit situaci např. jejich vypuštěním, doplněním apod. Konečně pokud data nejsou navržena či kódována na míru problému, např. pochází z jiného zdroje, nebo věcná analýza nebyla provedena důsledně, mohou obsahovat data irelevantní, s konstantními hodnotami, s nadbytečnou informací, nevhodně zakódovaná apod. Výsledkem etapy filtrování dat mají být data bezchybná, relevantní, zakódovaná v souladu s následným zpracováním a s jednotným zakódováním údajů chybějících, prostě data formálně i věcně správná, konzistentní.

5. **Předzpracováním dat** rozumíme některé další úpravy dat před jejich analyzováním, které si vyžadují následně použité metody z hlediska věcné správnosti. Patří sem různé transformace údajů jako standardizace atributů pro odstranění závislosti na jednotkách měření nebo normalizace pro odstranění závislosti dat na velikosti objektů, někdy i dichotomizace či kategorizace údajů, transponování matice dat (i když tyto úpravy mohou být řazeny k předcházející etapě), transformace souboru dat do nových souřadnic odpovídajících hlavním komponent apod. Tyto transformace obvykle přímo souvisí s následným použitím některé metody.

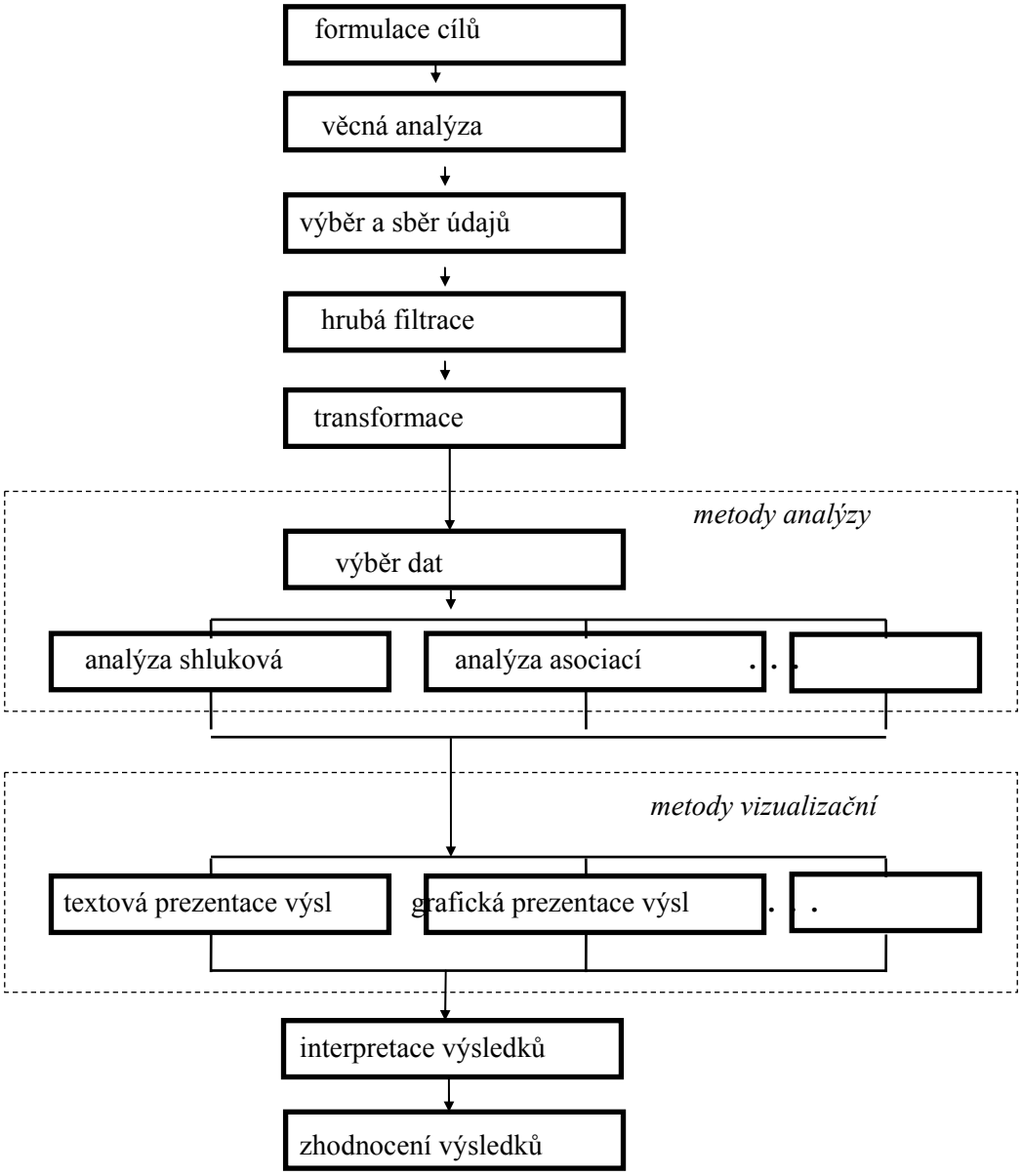
6. **Vlastní analýzy** dat obsahují řadu metod, jejichž výsledky jsou cílem celého procesu. Patří sem metody matematické statistiky a příbuzných disciplín (gnostika, fuzzy) a patří sem metody explorační analýzy. Pro konkrétní data se obvykle provádí řada výpočtů realizujících jednotlivé metody. Jejich výběr souvisí se zadáním z etapy problémové analýzy.

7. Prezentace výsledků sice nepřináší nové výsledky, ale jejich nové zobrazení, **vizualizace** dat a výsledku analýz může výrazně ulehčit jejich pochopení a následnou interpretaci. Výsledky výpočtů nad daty mohou mít různou formu. Nejjednodušší forma numerická, byť uspořádaná do sestav, tabulek apod. obvykle znamená pro odborníka ještě mnoho práce při „překladu“ do vlastní odborné

terminologie. Mnohem názornější jsou doplňující výstupy grafické nebo textové. Forma prezentace výsledků, jejich nové zobrazení může výrazně ulehčit jejich pochopení a následnou interpretaci. V systémech pro podporu rozhodování je na prezentaci výsledků kladen obzvláště velký důraz a všem uživatelům samozřejmě ušetří mnoho manuální práce.

8. **Interpretace výsledků.** Celá analýza nespočívá jen ve vlastních výpočtech nad daty. Aby byly výpočty užitečné, musí být provedena interpretace výsledků, jejich věcná slovní formulace. To je etapa, kterou obvykle provádí analytik společně s odborníkem, protože jde opět o rozhraní mezi jazykem matematiky a věcnou problematikou.

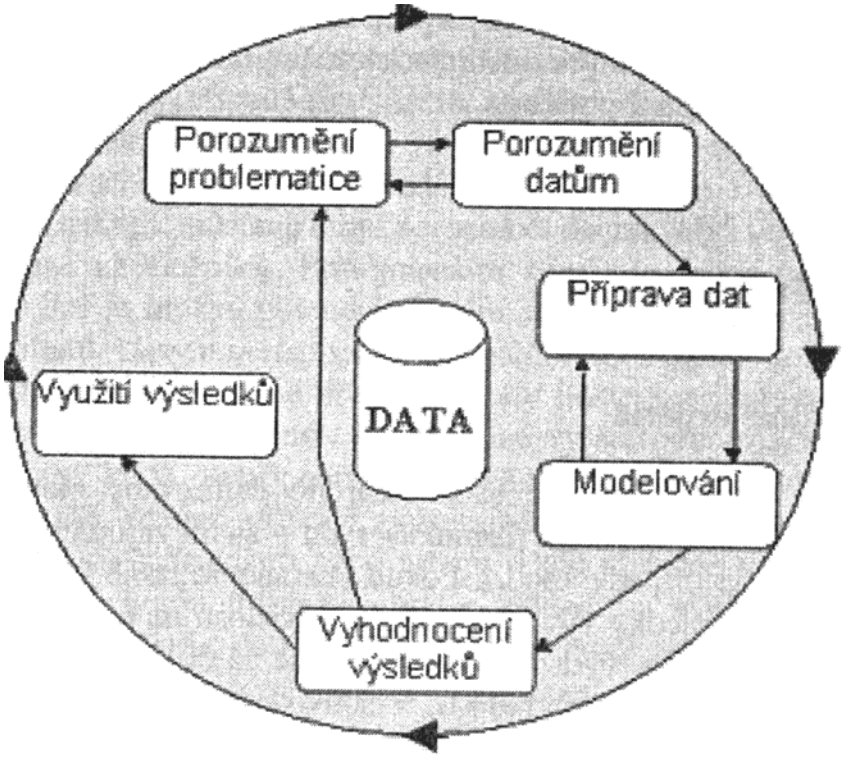
9. **Vyhodnocení průzkumu** je poslední fází, kterou provádí odborník, případně opět ve spolupráci s analytikem. Podle typu výzkumu vyhodnotí například, zda přinesl očekávanou kvalitu výsledků, zda se bude pokračovat v dalším sběru dat a dalších analýzách nebo jsme u konce výzkumu, zda jsou výsledky takové, že budou dále testovány metodami matematické statistiky, zda jsou výsledky natolik průkazné, že je možno formulovat zákonitosti (přírodní ap.) a přejít tak k vyšší fázi zkoumání.



Obrázek 1.4. Životní cyklus procesu získávání znalostí

Metodologie CRISP – DM

Často je používána jako standardní metodologie CRISP – DM (**CR**oss **I**ndustry **S**tandard **P**rocess model for **D**ata **M**ining). Jde opět o typ životního cyklu, neliší se v podstatě ničím od výše uvedeného postupu a je znázorňována takto:



Obrázek 1.5. Metodologie CRISP – DM

Metodologie SEMMA

Jednou z dalších metodologií často citovaných je SEMMA. Pomáhá snížit závislost na analytických expertech a zároveň slouží jako průvodce při implementaci data mining pro určitý problém.

Zkratka SEMMA znamená pět fází v data mining procesu:

Sampling – vzorkování, identifikování množiny vstupních dat

Exploring - prozkoumávání množiny dat statisticky a graficky

Modofocation – úprava dat, příprava na analýzu

Modeling – modelování, výběr vhodného prediktivního modelu

Assestment – zhodnocení, srovnání konkurenčních prediktivních modelů

Role analytika při dolování znalostí

Při cíleném výzkumu nebo při dolování z databází je nutno provést analýzu řešeného problému:

- **analýzu datovou** (jakou část reality popisují data, které údaje jsou či budou k dispozici pro analýzu, jakých syntaktických i sémantických typů jsou jednotlivé údaje, jejich rozdělení na dimenze pro agregování a fakty atd.),

- **analýzu problémovou** (jak je nutno data připravit – zakódovat, filtrovat, transformovat, než je možno použít konkrétních metod analýzy, jaké jsou požadovány typy výsledků, jaké typy metod jsou pro tato data použitelné)
- **analýzu forem prezentace výsledků**

❑ SW pro podporu dolování

V současném softwarovém světě je nabízena řada produktů, jejichž slogany zní velmi slibně jak z hlediska automatických postupů, tak uživatelského prostředí. Žádný z nich neslibuje dřinu, nutnost mnoho studovat, dlouho analyzovat, znovu a znovu analyzovat, probírat se množstvím banalit často krásně barevně provedených, 95% svých pracně získaných výsledků zahodit a jen při nemalém štěstí se dobrat malého množství skutečně nových znalostí. A to vše za ceny produktů desetitisícové, statisícové i milionové.

Podrobněji se o některých SW produktech zmíníme později.

Shrnutí pojmů 1.

Realita a její poznávání.

Statistika, analýzy mnohorozměrných dat, konfirmační a explorační analýzy dat.

Úlohy dolování znalostí z dat.

OLAP a dolování znalostí.

Využití dolování znalostí v praxi.

Typy uživatelů metod dolování a jejich potřeby.

Zdrojová data, použité metody pro různé oblasti praxe.

Využití dolování znalostí pro získání obchodních výhod, pro získání nových odborných znalostí, pro politické a sociální průzkumy.

Životní cyklus procesu získávání znalostí z dat. Etapy procesu získávání znalostí z dat

Metodologie CRISP – DM, metodologie SEMMA.

Role analytika při dolování znalostí.

Otázky 1.

1. Co je dolování znalostí z dat?
2. Jak souvisí metody dolování znalostí s metodami matematické statistiky?
3. Které metodologie pro dolování znalostí znáte a jak se liší?
4. Jaký je rozdíl mezi OLAP a dolováním znalostí?
5. Jaká je role analytika při dolování znalostí?

Semestrální projekt Analýza 1

Najděte si vlastní data pro analýzy.

Abychom měli na co aplikovat metody, se kterými se v MAD seznámíme, musíme si nějaká data opatřit. Možností, jak data získat, je několik:

1) Nejjednodušší možnost je najít a stáhnout nějaká data z internetu. Na internetu existují archívy dat, které se požívají pro testování dataminingových metod. Například poměrně známý archív UC Irvine Machine Learning Repository si lze prohlédnout na adrese http://archive.ics.uci.edu/ml/. Nevýhodou tohoto přístupu je, že data jsou již mnohokrát analyzována, a tudíž se toho o nich již mnoho nového prostřednictvím našich analýz nedozvíme. Další nevýhoda může nastat v případě, že se nedostaneme do kontaktu s autorem dat – pokud nám pak tedy v popisu dat o nich něco není jasné, jen obtížně se k vysvětlení dostáváme. *Pochopení dat a významů atributů v nich totiž hraje v celém analytickém procesu zásadní roli.* Pokud nechápeme, co vlastně analyzujeme (o čem přesně data jsou), tak **zaprvé** zřejmě povedeme celý analytický proces špatným směrem a **zadruhé**, což je ještě horší, buďto nebudeme vůbec schopni výsledky analýz interpretovat anebo je budeme interpretovat špatně, což je ta nejhorší možná varianta. Je proto dobré stáhnout si taková data, která buď budou velmi dobře a pochopitelně popsána v nějakém přiloženém souboru, nebo data, u kterých máme kontakt na jejich autora, který je ochoten k datům poskytnout nějaké dodatečné informace.
2) Druhou možností, jak získat data, je oslovit známé, kolegy či kamarády, kteří s nějakými daty pracují, a požádat je, aby nám dali data k dispozici. Tato varianta je poněkud náročnější (určitě nemáme takový výběr z mnoha souborů dat, jako na internetu, má však obrovskou výhodu v tom, že jsme v kontaktu s autorem dat, tudíž nejasnosti ohledně dat je možno s ním prokonzultovat. Další výhoda je ta, že výsledky analýz mohou být pro tuto osobu užitečné, což může být další motivace pro práci.
3) Třetí možností je udělat vlastní sběr dat, např. formou dotazníků (papírových či elektronických), v oblasti, která nás zajímá. Tato forma získání dat je nejpracnější, má ale samozřejmě své výhody – autorem dat jsme my, tudíž datům rozumíme a zřejmě nás zajímají i výsledky analýz, což nás motivuje k tomu, abychom analýzy provedli pořádně a získali nějaké seriózní výsledky.

Pokud jsme si tedy opatřili data kteroukoli formou, je potřeba se s nimi velmi dobře seznámit, zjistit případně již známé vztahy mezi atributy, popřípadě pokud existují, seznámit se s výsledky předchozích analýz nad těmito daty.

2. METODY PŘEDZPRACOVÁNÍ DAT

Čas ke studiu: 3 hodiny

Cíl Po prostudování této kapitoly budete umět

- rozdělovat data z hlediska možností budoucího použití dolovacích algoritmů
- filtrovat, transformovat a odvozovat data pro následné dolování
- pomocí výpočtu hlavních komponent data transformovat, nebo určit jejich charakteristické atributy

Výklad

2.1. Data pro analýzy

Data a jejich typy

Jak jsme uvedli, zkoumané objekty a jevy (z hlediska zákonitostí jejich chování a vztahů v realitě) popisujeme souhrnem jejich vlastností, atributů. Ze všech atributů objektu se pro zamýšlený výzkum vybírají ty, které s problémem souvisejí. Výběr relevantních atributů se provádí v rámci datové analýzy problému. Jde o úkol často velmi náročný a rozhodující pro kvalitu výsledku, ale pravděpodobně nealgoritmizovatelný. Expert na věcnou problematiku provede výběr samozřejmě efektivněji než laik. Při výzkumu dosud neznámé oblasti je vhodné volit v případě nejistoty o tom, zda atributy se zkoumanou skutečností souvisejí, raději atributů více.

Data pro analýzy můžeme dělit podle několika hledisek.

- **Z hlediska syntaktického** se dělí data podle standardních datových typů na numerická, textová, datumová a časová, logická, ostatní nestrukturované typy obvykle hromadně označované OLE (Object Linking and Embedding).

  V posledních desetiletích vznikly databáze nejen s klasickou strukturovanou informací (relační databáze), ale s uloženou informací mnoha dalších datových typů. Patří mezi ně například

  - transakční databáze obsahující data o obchodních transakcích, zahrnující mimo několik strukturovaných údajů o transakci dále seznam položek, které transakci tvoří; například seznam nakoupeného zboží při jednom nákupu (někdy jsou tato data nazývána nákupním košíkem),
  - objektově-relační databáze, které pracují s objekty s neatomickými atributy, jako seznamy, tabulkami atd.
  - textové databáze obsahující mimo strukturovanou informaci také nestrukturované rozsáhlé dokumenty,
  - prostorové databáze, například databáze geografických informačních systémů, obsahujících také data prostorová, jako zeměpisné souřadnice objektů,
  - temporální databáze obsahující také časové údaje a funkce pro práci s nimi,

- multimediální databáze obsahující data obrazová, audiosekvence, videosekvence, časové řady a další neatomické položky,
- heterogenní databáze obsahující všechny možné typy dat, například Web,
- a další.

Poznámka:

V tomto předmětu se budeme dále zabývat jen daty strukturovanými a vhodnou konverzí převeditelnými na data numerická (konverze viz níže). Rozvíjející se metody dolování z dat textových, obrazových, multimediálních a dalších zde probírat nebudeme. Často však tyto pokročilé metody dolování v první fázi převedou zkoumané údaje na data numerická a pak používají klasické metody dolování.

Pro dolování znalostí rozlišujeme údaje jemněji podle významu.

- **Z hlediska sémantického** dělíme dále data na

Numerické údaje

- **binární** (též dvouhodnotové, dichotomické, alternativní), nabývají pouze dvou hodnot

  Příklad 2.1. *{0,1}, {ano, ne}, {true, false}, {kuřák, nekuřák}, {muž, žena}, ...* ♦

- **kategoriální** (též kvalitativní, klasifikační, nominální), nabývají hodnot malého konečného počtu hodnot a znamenají příslušnost k jisté kategorii, udané svým očíslováním {0,1,...,k} bez významu kvantitativního, tedy bez uspořádání podle velikosti;

  Příklad 2.2. národnost $\in$*{česká,slovenská,polská,...} očíslována 1 = česká, 2 = slovenská, ...* ♦

- **ordinální** (též pořadové), nabývají také hodnot {0,1,...,k}, je u nich dáno přirozené uspořádání, případně bez významu vzdálenosti mezi hodnotami; obvykle se s nimi pracuje jako s kategoriálními, někdy jako s celočíselnými reálnými

  Příklad 2.3. známky *{1 až 5} ve škole jsou uspořádány, ale vzdálenosti mezi nimi obecně nejsou stejné, ...* ♦

- **reálné** (též reálněhodnotové, intervalové, kvantitativní), nabývají reálných hodnot z intervalu <a,b>, jsou zaznamenány s danou přesností; hodnota má absolutní význam v daných jednotkách; z hlediska významového se nerozlišuje, zda je přesnost na 0 desetinných míst a jsou tudíž celočíselná, nebo mají desetinnou část;

  někdy se uvádí jako samostatný typ údaje **poměrové**, obvykle udávající podíl dvou absolutních údajů; nabývají opět hodnot z <a,b> zaznamenané s danou přesností; hodnota má význam relativní, jejich stupnice není lineární; rozlišení s reálnými daty se projeví pouze při interpretaci výsledku.

  Příklad 2.4. *reálné* věk, výška, váha, cena ..., *poměrové* .... procento daně ♦

Nenumerické údaje

- **časové**, vyjadřující absolutní datum nebo čas události, případně časový úsek; používají se obvykle k přepočtu na časový úsek (údaj reálný) nebo pro kategorizaci.

  Příklad 2.5. *časové údaje* datum narození, termín splatnosti, začátek výukové hodiny, okamžik startu, *nebo doba události* délka letu, doba běhu na 100m, počet dnů školení *apod.* ♦

- **textové**, vyjadřující slovně nebo znakovým kódem popisovanou vlastnost; podle okolností se používají k výběru podmnožin objektů, je-li to možné, tak se číselně zakódují a zpracovávají jako data kategoriální či ordinální, případně se zpracovávají dalšími metodami.

Příklad 2.6. *typické texty bez možnosti jednoduchého zakódování jako* anotace článku, text článku, poznámka o čemkoliv, *..., ale případně také snadno zakódovatelné texty jako* rodinný stav *{svobodný, ženatý, ...}, nebo jinak upravitelné údaje jako* rodné číslo *(lomítko se zruší nebo zapíše jako desetinná tečka) apod.* ♦

- **grafické, zvukové, ostatní neatomické údaje,** obecně označované **OLE**; pokud se účastní zpracování, předzpracují se obvykle náročnějšími metodami, které jsou obsahem samostatných disciplín a kódují se na některý z předchozích typů.

  Příklad 2.7. *záznam EKG, záznam hudby, foto, videozáznam, ...* ♦

Většina metod analýzy mnohorozměrných dat pracuje pouze s numerickými údaji a před zpracováním se ostatní datové typy vhodně kódují. Způsob kódování je součástí etapy analýzy a souvisí s potřebami úlohy a budeme je probírat v rámci transformací údajů.

2.2. Metody filtrace a integrace dat

❑ Zdroje dat

Data vstupující do zpracování mohou pocházet z různých typů zdrojů. To může mít vliv na způsob jejich dalšího zpracování.

Buď jsou dobře navržena, zakódována a sesbírána přímo pro potřeby tohoto průzkumu, pak se v nich mohou objevit jen chybné údaje způsobené chybami při jejich měření, sběru, záznamu na médium počítače.

Druhou možností je, že byla data pořízena bez předchozí analýzy a obsahují údaje sice související s problémem, ale nejsou dosud ve formátu vhodném pro analýzy. Prakticky všechny dále uváděné metody vyžadují data numerická.

V praxi se vyskytující datové typy, časové, datumové, textové, grafické či zvukové, pokud se mají účastnit dalšího zpracování a nebýt jen informativním doplňkem pro zkoumané objekty, je nutno pro ně dodatečnou datovou analýzou navrhnout způsoby, jak z nich získat údaje numerické. Nebo jsou data vzniklá pro jiný účel a dodatečně zvolena pro analýzu. Pak je potřeba navíc vybrat údaje pro zamýšlené zpracování relevantní a z formálního hlediska vhodná.

Pokud data získává nějaký automat, například pomocí čidel, chápeme to jako speciální případ první možnosti a data pak nejsou zatížena lidskými chybami.

❑ Filtrace dat

Filtrací dat nazýváme řadu akcí, vedoucích k získání dat připravených ke spuštění dolovacích algoritmů. Je to činnost jen zdánlivě jednoduchá, skoro vždy však spotřebuje mnohem více času a práce, než vlastní dolovací výpočty.

Pro informovanější rozhodování je vhodné provést nejprve výpočty základních statistických charakteristik každého atributu.

Filtrace pak zahrnují

- výběr atributů vhodných k analýzám
- ošetření nebo vyloučení dat chybných, chybějících, redundandních, irelevantních, konstantních
- sjednocení formátů, měrných jednotek
- numerické zakódování některých dat, sjednocení kódování, kategorizace a dichotomizace dat; některé z těchto úprav mohou být řazeny již k transformacím dat.

Výsledkem filtrace mají být data numerická, formálně i věcně správná, konzistentní. Numerické atributy budeme rozlišovat na reálné, ordinální a kategoriální. Některé atributy se mohou opakovat v různých podobách (např. reálný věk a věk kategorizovaný do tříd, viz též odvozená data) pro různé typy metod dolování.

Jednotlivé metody filtrací:

- **Integrace dat, sjednocení formátů, měrných jednotek**

  je první operací při získávání dat z různých zdrojů. Musí se navrhnout současně s výběrem dat a na míru zdrojovým datům. Operace pro integrování dat je vhodné uložit do metadat pro pozdější opakující se načítání přírůstků dat nebo pro opakované zpracování na jiném vzorku dat.

- **Výběr relevantních dat pro analýzy – tzv. volba modelu**

  výběr dat, která se mohou používat pro různé typy dolovacích metod; výběr může být pro každou metodu poněkud jiný, výběr se provádí především pro atributy, někdy i pro objekty. Obvykle se tento proces nazývá tvorbou modelu pro dolování.

  Dle zvolené metody rozdělení vybraných atributů podle významu na

  dimenze, fakty (viz datové sklady)
  antecedenty, sukcedenty (předpoklady a následky, viz asociace, rozhodovací stromy)
  ovlivnitelné, neovlivnitelné (viz asociace i jiné metody)

- **Statistické charakteristiky atributů**

  Výpočet charakteristických hodnot atributů - průměr, minimum a maximum, standardní odchylka, medián, četnosti výskytů jednotlivých hodnot (frekvencí), počty chybějících údajů.

  Je užitečné tyto statistické charakteristiky zařadit k metadatům a uschovat. Jsou využívány u mnoha metod dolování. V etapě předzpracování jsou velmi užitečné k prvnímu „náhledu“ na data pomocí vizualizačních metod i k odhalení některých chyb v datech.

- **Statistické charakteristiky dvojic atributů**

  některé statistické charakteristiky se již používají pro analýzy, například

  korelační koeficienty
  regresní křivky
  frekvenční (kontigenční) tabulky

Pro data X a jejich kategoriální veličiny A1 a A2 má frekvenční tabulka tvar

A1 \ A2	0	1	2	...	k2	
0	a_{00}	a_{0}			a_{0h2}	a_{0}
1	a_{10}	a_{11}			a_{1h2}	a_{1}
2						
...						
k1	a_{h10}	a_{h11}	a_{h12}		a_{h1h2}	$a_{h1.}$
	$a_{.0}$	$a_{.1}$	$a_{.2}$		$a_{.h2}$	m

- **Odhalení dat chybných**
  - pro záznam dat do databáze použít vstupní program s kontrolami syntaktickými i logickými, ne textový editor,
  - použít kontrolní funkce při prvotním načítání dat do DM systému,
  - opakovaná kontrola celého procesu přípravy dat,
  - využití statistických charakteristik – například minima a maxima jednotlivých atributů porovnat se zadanými doménami atributů.
- **Řešení dat chybějících**
  - statistiky odhalí chybějící údaje, důležité jednotné označení v celém souboru a ve shodě s potřebami analýz,
  - pokud metody neumí zpracovat soubor s chybějícími údaji a data neúplná jsou, je nutné situaci řešit; pohodlnou možností je vyloučit taková data ze zpracování - celý atribut, objekty; soubor se zmenší,
  - jinou možností je doplnit chybějící údaje „vymyšlenými“ hodnotami; strategie optimistická (maximálními hodnotami atributu), pesimistická (minimálními), neutrální (průměrnými), jiná; soubor dat zůstane celý, ovšem s jistou nepřesností,
  - některé metod umí zpracovat nebo eliminovat chybějící data a stačí, když jsou správně označena.

Příklad 2.8.

Jsou dána data o pacientech interního nemocničního oddělení, obsahující údaje:

Pacient (pohlavi {muž, žena},
vek [rok],
stav {svobodny, zenaty, rozvedeny, vdovec},
prijem [datum příjmu],
puls [počet],
tlak [horni, dolni],
zlozvyky {nic, kouří, pije, kouří i pije},
diagnoza [nemoc],
vysledek {propuštěn domů, přemístěn na jiné odděl, zemřel})

Navrhněte metody předzpracování těchto dat.

Řešení: atributy **pohlavi, stav, zlozvyky** *a* **vysledek** *se budou jednoduše kategorizovat = kódovat na 0 / 1 a 0 / 1 / 2 / 3 / 4,*

vek *bude vhodné kategorizovat po konzultaci s lékařem do nového atributu* **vek_k**, *například do intervalů 1-3,4-6,7-10,11-14,15-17,18-25,26-35,36-50,51-60,*

z atributu **prijem** *se odvodí několik atributů* den, mesic, rok, …

diagnóze *se po konzultaci s lékařem přidělí numerický kód, odpovídající nemoci a skupinám nemocí, případně se použije celostátní číselník diagnóz.*

♦

2.3. Metody transformací dat

Data filtrovaná nemusí být ještě vhodně připravena pro všechny metody dolování. Například při shlukování se často měří podobnost objektů Eukleidovskou vzdáleností a v případě atributů různých měrných jednotek a jejich řádově rozdílných hodnot by vliv některých atributů silně negativně ovlivnil výsledek. Proto je někdy nutné odstranit vlivy měrných jednotek, jindy odstranit vliv velikosti objektů.

K transformacím dat řadíme jednak převádění dat mezi datovými typy – pokud to použité metody vyžadují, jednak složitější transformace, řešící například problémy se vzájemnou porovnatelností informací.

❑ Převody datových typů

Různé metody dolování

- **Kategorizace (diskretizace) reálných dat**
  - kategorizace reálných údajů – ekvidistantní intervaly, dle rozložení četností v intervalech; kategorie se očíslují a data se transformují; ruční možnost nepravidelného rozložení kategorií dle konkrétní situace.
  - automatická kategorizace, je-li to vhodné vzhledem k daným hodnotám atributu, volí se počet kategorií
- **Kategorizace nenumerických dat**
  - textových, datumových, časových, dalších,
    pokud atribut nabývá jen malého počtu hodnot, je možno okódovat hodnoty, uložit je do číselníku (metadat) a dále pracovat s numerickými kategoriálními nebo ordinálními údaji.
- **Dichotomizace dat**
  - atribut kategoriální o **k** kategoriích převést na **k** atributů s hodnotami z {0,1} - pro každou kategorii znamená jedna nová hodnota údaj {nabývá, nenabývá}.
  - pomocí výrazu [h1, . . ., hc], kde $hi \in \{0,1,2, . . ., k\}$, $hi \neq hj$ pro $i \neq j$. Výraz

    ❑ **Oi (Aj) [h1, . . ., hc]**

    znamená, že Oi (Aj) nabývá nebo nenabývá jedné z hodnot h1, ..., hc. Pak můžeme z kategoriální Aj vytvořit binární Ak podle pravidla Oi (Aj) [h1, . . ., hc] = 1 právě, když Oi (Aj) nabývá jedné z hodnot h1, . . ., hc, jinak Oi (Aj) [h1, . . ., hc] = 0

Příklad 2.9.

V datech o pacientech jsou mj. atributy věk, národnost*, ... Pokud budeme data zpracovávat metodou pro reálné atributy, můžeme věk použít beze změn,* národnost *(kategoriální) použít nemůžeme. Pokud použijeme metodu pro kategoriální data, můžeme atribut* věk *kategorizovat. Při datové analýze navrhneme způsob kategorizace, zde pro pacienty například do zadaných intervalů, které nemusí být pravidelné ani stejně četné, ale mají smysl ze zdravotního hlediska:*

věk < 10	věk_k = 1
věk ∈ <11,15>	2
věk ∈ <15,18>	3
věk ∈ <18,26>	4
věk ∈ <27,35>	5

atd.

Pokud metoda vyžaduje data binární a chceme pomocí ní zpracovávat i věk, *můžeme kategorizovanou hodnotu* věk_k *převést na několik binárních atributů* věk_b_1, věk_b_2, ..., *kde každá hodnota nabývá hodnoty 0 nebo 1 podle toho, zda věk patří nebo ne do příslušné kategorie (intervalu).*

Jiný způsob dichotomizace můžeme podle potřeby použít například seskupováním kategorií:

věk_k ∈ <1, 2, 3>	věk_b_1 = 1, jinak věk_b_1 = 0
věk_k ∈ <4, 5,6>	věk_b_2 = 1, jinak věk_b_2 = 0
věk_k ∈ <7, 8, 9>	věk_b_3 = 1, jinak věk_b_3 = 0

♦

❑ Transformace dat

Data vhodných datových typů, bezchybná, formálně správná, nemusí být ještě vhodným vstupem pro všechny typy analytických metod.

- **Normalizace objektů**

Normalizací objektů s reálnými atributy rozumíme odstranění závislosti dat na velikosti objektů; objekty se normalizují podle transformačního vztahu (v matici dat **X** je původní hodnota j-tého atributu u i-tého objektu označena x_{ij}, přepočtená hodnota z_{ij}

$$z_{ij} = \frac{x_{ij}}{\sqrt{\sum_{j=1}^{n} x_{ij}^2}} = \frac{x_{ij}}{P}$$

kde P je norma objektu.

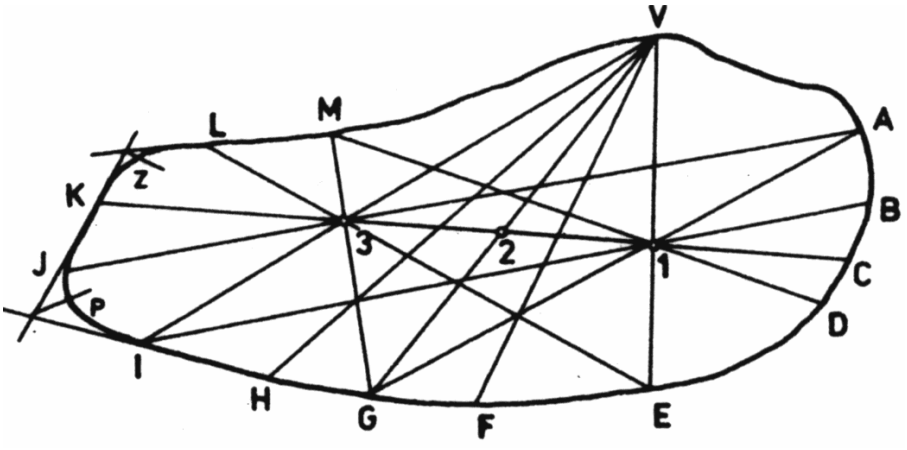
X

	A1	**A2**	**…**	**Aj**	**…**	**An**
O1	x11	x12		…	…	
	…	…				
Oi	xi1	xi2				
	…	…				
Om	xm1	xm2				

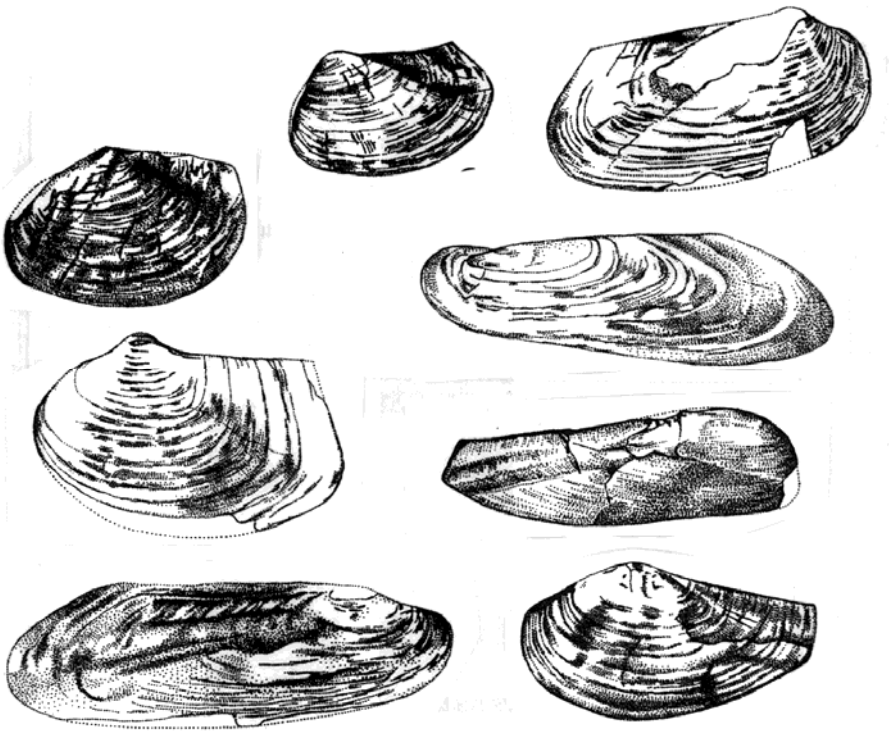
Příklad 2.10. [11]

Je dána matice dat, kde objekty jsou mlži několika rodů, popsané 25 rozměry (v mm) dle obrázku. Hledají se tvarově blízké množiny objektů.

Ke stejnému rodu zřejmě patří tvarově stejní, ale velikostí rozdílní mlži. Pokud bychom nechali absolutní rozměry v mm pro jejich popis a shlukování (hledání skupin podobných), zřejmě by velikost měla zkreslující vliv. Proto je nutné před shlukováním objekty – mlže normalizovat.



Obrázek 2.1. 25 rozměrů mlže popisujících tvar



Obrázek 2.2. Několik ukázek nalezených mlžů

♦

- **Standardizace atributů**

Standardizací reálného znaku rozumíme odstranění závislosti jednotlivých atributů na jednotkách měření. To je nutné prakticky u všech atributů s reálnými hodnotami. Metody založené na pojmu vzdálenosti, použité bez předchozí standardizace, by dávaly výsledky zkreslené vlivem rozdílných jednotek. Výsledná hodnota atributu po standardizaci je zhruba v intervalu <-1,1>.

X

	A1	A2	…	Aj	…	An	Aj_s
O1	x11	x12		x1j	…		z1j
	…	…		…			…
Oi	xi1	xi2		xij			zij
	…	…		…			…
Om	xm1	xm2		xmj			zmj

Atributy (sloupce) se normalizují podle transformačního vztahu

$$z_{ij} = \frac{x_{ij} - \bar{x}_j}{s_j}$$

$$\bar{x}_j = \frac{\sum_{i=1}^{m} x_{ij}}{m} \qquad s_j = \sqrt{\frac{\sum_{i=1}^{m} (x_{ij} - \bar{x}_j)^2}{m}}$$

Příklad 2.11.

Jsou dány údaje různých druhů o sledovaných studentech:

Student (věk, výška, váha, známka_MAT, známka_DEJ, ..., skok_vys, skok_dal, beh_100m, ...)

Úkolem může být i zkoumání, jestli nějak spolu souvisí tělesné a duševní výkony. Pak intervaly známek <1,5> a skoků <0, 800> by byly nesouměřitelné, absolutní hodnoty skoků v cm by měly větší „váhu" než známky. Po standardizaci bude váha obou stejná.

♦

- **Transponování matice dat**

Při hledání optimální podmnožiny atributů, charakterizujících daný objekt je možno použít shlukování vlastností, tedy původních atributů. K tomu je třeba matici dat předem transponovat, zvolit za „objekty" původní aspekty. Výsledky metod analýzy dostanou nový význam.

Příklad 2.12.

U shlukovacích metod si uvedeme příklad hledání skupin podobných lidí na základě odpovědí na otázky v dotazníku. To provedeme klasickým použitím shlukovací metody.

Když se ale datová matice transponuje, shlukováním dostaneme skupiny podobných si vlastností. Je to jiný výsledek, dává jinou informaci.

♦

- **Odstranění lineárních závislostí mezi atributy**

Může se stát, že v datech některé atributy jsou vzájemně závislé, přičemž tato závislost není předem známa. Výsledky tak mohou zkreslené a je vhodné takové závislosti před dolováním odhalit a korigovat. Tímto problémem se zabývá **metoda hlavních komponent.**

Protože jde o metodu, která se řadí jak k transformačním metodám předzpracování dat, tak k metodám analýzy dat, věnujeme jí podrobně následující kapitolu.

2.4. Odvozování dat

❑ Generování odvozených atomických údajů

U úloh dolování je často užitečné z existujících atributů pro analýzy odvodit nové atributy, byť jsou jejich údaje redundandní. Je sice pravda, že v případě jejich potřeby je možné je vždy znovu vypočítat a nerozšiřovat tak velikost databáze, ale předpočítané údaje jsou pro využití pohodlnější a hlavně nezdržují časově náročné dolovací algoritmy.

Neexistuje zde jednotný návod k použití, záleží na potřebách analýz a nápaditosti analytika. Je velmi užitečné, když používaný SW má k dispozici nástroje pro zadání vzorců pro výpočet odvozených veličin, případně zabudovány funkce pro některé standardní často používaná odvození.

Příklad 2.13.

Datumové a časové položky (které v databázích mají pravidelně své datové typy) se samy o sobě metodami dolování zpracovávat nedají (nejsou to numerické údaje), ale jsou bezpochyby pro analýzy potřebné. Proto se z nich odvozují mnohé numerické údaje, jako den, měsíc, rok, den v týdnu, roční období, znamení zvěrokruhu, …, *nebo ze dvojice časových atributů délku období apod.*

Klasická odvození jsou věk, pohlaví, datum narození z rodného čísla, počet dnů, měsíců, roků *ze dvou datumových údajů*, *kategorizace* rodinného stavu slovně vyjádřeného apod.

♦

❑ Agregace údajů

Zvláštním případem odvozených dat jsou data agregovaná:

sumy, průměry, minima a maxima, počty případů, relativní četnosti apod.

Z nahromaděných dat za různá časová období, za různé geografické celky či jiné dimenze jsou tyto údaje bezpochyby užitečné samy o sobě a někdy jejich grafické či tabulkové prezentace bývají vydávány i za výsledky dolování.

V dalších kapitolách se seznámíme s tím, jak agregované údaje mohou využívat některé dolovací algoritmy.

Jinak (jak již víme) jsou agregované údaje, i hierarchicky uspořádané, hlavním zdrojem výsledků, které produkuje OLAP se svými prezentačními metodami.

Příklad 2.14.

Údaje o pacientech a jejich nemocech dlouhodobě sbírané mohou sloužit k vědeckým a statistickým účelům. Nebudou pak základním objektem zájmu jednotlivé případy nemocí, ale

- *počty případů nemocí po dnech (při epidemii), po týdnech a měsících či rocích, v jednotlivých okresech, krajích, zemích apod.,*
- *průměrné dávky léků podávané jednotlivým věkovým, profesním a jiným skupinám pacientů,*
- *součty množství spotřebovaných léků podle druhů, zemí,*
- *minimální a maximální dávky nových léků podávaných testované skupině pacientů*
- *atd.*

♦

Shrnutí pojmů 2.

Zdroje dat pro analýzy.
Datové typy syntaktické.
Datové typy sémantické.
Integrace dat, sjednocení významu atributů, formátů a měrných jednotek.
Metody filtrace dat.
Numerické zakódování údajů.
Metody transformací dat.
Převody datových typů. Kategorizace, dichotomizace.
Normalizace a standardizace. Transponování matice dat.
Odvozování dat.
Generování odvozených atomických údajů.
Agregace údajů a jejich využití.

Otázky 2.

1. Co všechno zahrnuje etapa předzpracování dat?
2. Jak rozlišujeme z hlediska sémantiky údaje pro zpracování a jak numerické údaje?
3. Co znamená integrace dat a proč se provádí?
4. Z jakých hledisek je potřeba zkontrolovat data a provést operace s daty při jejich filtraci?
5. Proč se převádí některé sémantické datové typy na jiné?
6. Jak se provádí převody datových typů?
7. Co znamenají transformace dat a které transformace znáte?
8. Co je standardizace, pro jaká data a proč se provádí?
9. Co je normalizace, pro jaká data a proč se provádí?
10. Co jsou hlavní komponenty pro datovou matici?
11. Kdy se používá výpočet hlavních komponent a k čemu?
12. Jaký význam pro dolování mají odvozené údaje?
13. Jaké typy odvozených údajů znáte?

Úlohy k řešení 2.

1. Jsou dána data z evidence sportovního gymnázia s atributy: školní rok, třída, učitel [jméno], jméno [studenta], pohlaví [chl / dív], věk, výška, váha, dále maximální výkony sportovní za tento rok ve skoku vysokém [cm], dalekém [cm], běhu -100m [12.3 sec], běhu - 400m a závěrečné známky z češtiny, cizího jazyka [ruština a později angličtina], matematiky, fyziky, dějepisu a zeměpisu. Data jsou pořizována za dobu 30 let.

   Navrhněte úplné předzpracování těchto dat.

2. Jsou dána data BANKA, obsahující 1027 záznamů za minulých 5 let. Jejich struktura je

Banka (pohlavi [m / z], vek [roků], svobodny [a / n], nezamestnany [a / n], cim_ruci [auto, dum, PC, kolo], problematicky_region [a / n], mes_prijem [Kc], hotovost_u_banky [Kc], pocet_mesicu_splatky, pocet_roku_u_souc_firmy, dostal_uver [a / n], splacel_bezproblemove [a/n])

Navrhněte úplné předzpracování těchto dat.

3. Jsou dána data Pacienti s atributy infarkt [ano / ne], ang_pectoris [ano / ne], berc_vred [ano / ne], vaha [kg], vyska [cm], kurak [ano / ne], pohlavi [muž / žena], vek [roků], město [ano / ne], duchodce [ano / ne], stres [ano / ne].

   Navrhněte úplné předzpracování těchto dat.

4. Data pocházejí ze sociologického výzkumu (ankety). Tato anketa se týká vědního oboru, který se nazývá antroponymie. Je to vlastně součást onomastiky (onomatologie), zkoumající vlastní jména živých bytostí. Pro informaci onomastika se dále dělí na toponomastiku (zkoumá vlastní jména neživých věcí) a chrématonomastiku, zkoumající vlastní jména lidských výtvorů a zařízení.

   Anketa byla prováděna v několika základních školách s polským jazykem vyučovacím v pohraničí (polsko - českém), a jejími účastníky byli nejen žáci těchto škol, ale i jejich rodiče. Cílem ankety bylo popsat antroponymické struktury ve školním i mimoškolním prostředí. Co všechno se pod tímto pojmem skrývá, nám nejlíp popíše výčet jednotlivých otázek pro žáky“

   1. Křestní jméno, příjmení
   2. Věk
   3. Základní škola v: Třída
   4. Bydliště: okres: Město/Vesnice
   5. Máš sourozence? Počet:
   6. Jakou formu tvého jména používal (používá)
      a) matka (v předškolním věku, současně)
      b) otec (v předškolním věku, současně)
      c) babička
      d) dědeček
      e) sourozenci
      f) kamarádky
      g) kamarádi
      h) učitelé
   7. Znáš přezdívku týkající se
      a) jména tvého, kamarádky, kamaráda, učitele
      b) příjmení tvého, kamarádky, kamaráda, učitele
   8. Znáš říkadla týkající se tvého jména, jména kamarádek, kamarádů?
   9. Jaké formy jmen sourozenců, kamarádů, kamarádek používáš?
   10. Jaké formy jmen používají rodiče při vzájemném oslovování?
   11. Jakým způsobem se nejčastěji oslovují kamarádi, kamarádky u vás ve třídě (jméno, příjmení, přezdívka, nevím)?
   12. Jakým způsobem učitelé nejčastěji oslovují žáky u vás ve třídě (jméno, příjmení, přezdívka, nevím)?
   13. Jsi spokojen se svým jménem?

Semestrální projekt Analýza 2

Pro svá data navrhněte úplné předzpracování.

3. HLAVNÍ KOMPONENTY

Cíl Po prostudování této kapitoly budete vědět

- co jsou latentní proměnné
- jak se dají vypočítat
- k čemu slouží výsledky výpočtu hlavních komponent
- kdo transformaci do hlavních komponent navrhuje a na základě čeho

Výklad

❑ Skryté proměnné

Mezi transformační metody je možno zařadit i metodu hlavních komponent, i když její výsledky dávají i užitečné informace o datech. Jde o součást lineární faktorové analýzy, patřící mezi tzv. modely s latentními proměnnými.

Obsahuje-li datová matice závislé atributy, mohou být některé analýzy zkreslené. Metoda vychází z předpokladu, že dvě proměnné mohou být závislé proto, že obě měří tutéž skrytou společnou veličinu, nazývanou společný faktor nebo hlavní komponenta.

Příklad 3.1.

neměřitelná veličina = diagnóza,
měřitelné atributy = teplota, tlak, nález v krku, ...

nebo

neměřitelná veličina = nadání matematicko-logické
měřitelné atributy = známka z matematiky, známka z fyziky, ...

♦

Hlavních komponent může být v datech více a lze je matematicky zkonstruovat.

❑ Vzájemná závislost a nezávislost znaků

Některé znaky (atributy) lze považovat za náhodné veličiny ve smyslu matematické statistiky. Pak můžeme při posuzování jejich vzájemné závislosti používat prostředků matematické statistiky.

Ideálně by měly být zkoumané objekty charakterizovány atributy vzájemně nezávislými. Protože vzájemná závislost atributů může ovlivnit výsledky analýzy, je vhodné takové závislosti předem vyloučit, nebo alespoň o nich vědět a kvantitativně je ohodnotit.

Závislost atributů může mít nejrůznější charakter a obecný postup pro její nalezení neexistuje. Pokud však předpokládáme závislost alespoň přibližně lineární, lze použít pro její vyjádření koeficient kovariance nebo koeficient korelace.

Kovariance je míra vzájemné závislosti mezi dvěma náhodnými veličinami. Protože je její hodnota závislá na hodnotách zkoumaných atributů, používá se častěji tato hodnota standardizovaná - **koeficient korelace**. Ten pro dva atributy Ai, Aj nabývá hodnot z intervalu < -1, 1>, přičemž pro

hodnoty blízké nule považujeme atributy Ai, Aj za nezávislé, pro hodnoty blízké -1 nebo 1 považujeme oba atributy za lineárně závislé.

Pro matici dat X s reálnými atributy Ai, Aj

X

	A1	**A2**	**Ai**	**…**	**Aj**	**An**
O1	x11	x12	x1i	…	x1j	
	…	…			…	
Oi	xi1	xi2	xii		xij	
	…	…			…	
Om	xm1	xm2	xmi		xmj	

je definována kovariance $\mathbf{k_{ij}}$ a korelace $\mathbf{r_{ij}}$ vztahy

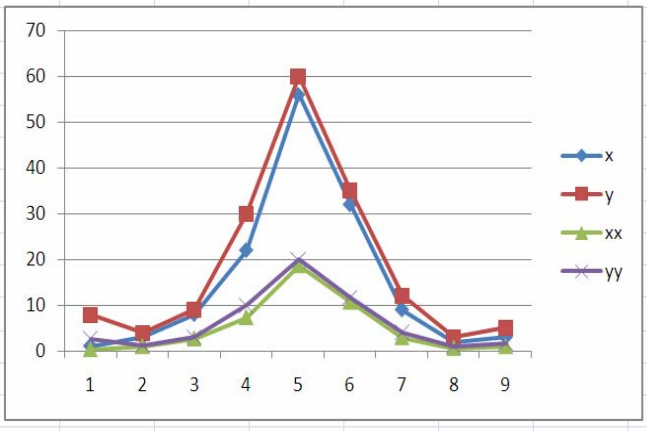
$$k_{ij} = \frac{1}{n}\sum_{l=1}^{n} x_{li} \,.\, x_{lj} - \bar{x}_i \,.\, \bar{x}_j \qquad\qquad r_{ij} = \frac{k_{ij}}{s_i \,.\, s_j}$$

kde x_{ij} je původní hodnota j-tého atributu u i-tého objektu v matici dat **X**,
$\underline{x}_i$ je střední hodnota i-tého atributu,
s_i je standardní odchylka i-tého atributu.
k_{ij} je hodnota kovariance i-tého a j-tého atributu
r_{ij} je hodnota korelace i-tého a j-tého atributu.

Příklad 3.2.

Jsou dány atributy x, y, platí xx = x/3, yy = y/3. Závislost obou dvojic atributů (x, y), (xx, yy) je tedy stejná.

x	y	xx	yy
1	8	0,333333333	2,666667
3	4	1	1,333333
8	9	2,666666667	3
22	30	7,333333333	10
56	60	18,66666667	20
32	35	10,66666667	11,66667
9	12	3	4
2	3	0,666666667	1
3	5	1	1,666667
covar(x,y)=	316,951	corel(x,y)=	0,992
covar(xx,yy)=	37,316	corel(xx,yy)=	0,992



Kovariance dvojic (x, y) a (xx, yy) jsou různé, závisejí na absolutních hodnotách atributů, standardizované korelace jsou shodné. Na grafu vidíme vysokou závislost obou dvojic.

♦

Korelační matice reálných atributů

Pokud vypočteme koeficienty korelace pro všechny dvojice atributů a sestavíme je do symetrické čtvercové matice **R** typu n x n, dostaneme **korelační matici**.

R

	A1	**A2**	**Ai**	**…**	**Aj**	**…**	**An**
A1	1	r12	r1i	…	r1j		r1n
…	…	…			…		
Ai	ri1	ri2	1		**rij**		rán
…	…	…			…		
Aj	rj1				1		rjn
…							
An	rn1	rn2	rni		rnj		1

Vlastní čísla a vlastní vektory matice korelační

Pro výpočet hlavních komponent potřebujeme nejprve vypočítat vlastní čísla a vlastní vektory matice **R** (K). Zopakujeme si proto některé pojmy z maticového počtu.

Charakteristickým mnohočlenem čtvercové matice **R** nazveme determinant matice $|\mathbf{R} - \lambda \mathbf{E}|$, kde **E** je jednotková matice řádu n. Jeho kořeny λ_i nazýváme vlastními (charakteristickými) čísly matice **R**.

Jde o matici symetrickou (protože $r_{ij} = r_{ij}$) s reálnými prvky. Platí, že vlastní čísla takové matice jsou reálná čísla. Protože jde o matici řádu n, je determinant matice $|\mathbf{R} - \lambda \mathbf{E}|$ mnohočlenem stupně n a charakteristická rovnice má n reálných kořenů. Bez újmy na obecnosti si můžeme kořeny uspořádat sestupně podle velikosti.

Stopou čtvercové matice R nazýváme součet diagonálních prvků matice $r_{11} + r_{22} + \ldots + r_{nn}$.

Vlastní čísla jsou tedy řešením charakteristické rovnice.

$$| \mathbf{R} - \lambda \mathbf{E} | = 0$$

kde λ_i, pro i = 1, . . ., n (kořeny této rovnice) jsou vlastní čísla matice R.

E je jednotková matice.

Jde tedy o rovnici

$$\begin{vmatrix} \mathbf{1-\lambda} & r12 & r1i & \ldots & r1j & & r1n \\ \ldots & \ldots & & & \ldots & & \\ ri1 & ri2 & \mathbf{1-\lambda} & & \mathbf{rij} & & rin \\ \ldots & \ldots & & & \ldots & & \\ & & & & \mathbf{1-\lambda} & & \\ rn1 & rn2 & rni & & rnj & & \mathbf{1-\lambda} \end{vmatrix} = \mathbf{0}$$

Jejím řešením jsou (sestupně dle velikosti uspořádaná) vlastní čísla $\boldsymbol{\lambda_1, \lambda_2, \ldots, \lambda_n}$.

Vlastní vektory $\mathbf{u_i}$ matice **R** dostaneme pomocí vlastních čísel korelační matice. Pro každé λi se odpovídající vlastní vektor $\mathbf{u_i}$ (i-tý sloupec **U**) vypočte z rovnice

$$(\mathbf{R} - \lambda_i \mathbf{E})\, \mathbf{u_i} = \mathbf{o}$$

kde **o** je nulový vektor.

Pro n vlastních čísel dostaneme n vlastních vektorů (sloupců) a tedy čtvercovou matici vlastních vektorů **U**.

❑ Výpočet vlastních čísel a vlastních vektorů

Metod výpočtu vlastních čísel a vektorů existuje v numerické matematice mnoho, pro symetrické matice existují speciální metody. Jednou z nich je například Jacobiho metoda, založená na maticové redukci.

Výpočet vlastních čísel a vlastních vektorů matice **R (K)** metodou Jacobiho:

Na vstupu je výchozí matice **R** a pomocná matice **U**, která je jednotková; pak se provádí eliminace **R** do diagonální matice a současně se paralelně transformuje stejnými operacemi pomocná jednotková matice **U.**

Před eliminací:

R

$$\begin{pmatrix} \mathbf{1} & \dots & \mathbf{r_{1i}} & \dots & \mathbf{r_{1n}} \\ & & & & \\ \mathbf{r_{i1}} & & \mathbf{1} & & \mathbf{r_{in}} \\ \dots & & & & \\ \mathbf{r_{in}} & \dots & \mathbf{r_{ni}} & & \mathbf{1} \end{pmatrix}$$

E

$$\begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & & & \\ \dots & & \dots & \dots & \dots \\ 0 & & & 1 & 0 \\ 0 & & & 0 & 1 \end{pmatrix}$$

Po eliminaci:

eliminovaná **R**

$$\begin{pmatrix} \boldsymbol{\lambda 1} & \dots & 0 & \dots & 0 \\ \dots & & & & \dots \\ 0 & \dots & \boldsymbol{\lambda i} & \dots & 0 \\ \dots & & & & \dots \\ 0 & \dots & 0 & \dots & \boldsymbol{\lambda n} \end{pmatrix}$$

eliminovaná E = **U**

$$\begin{pmatrix} \mathbf{u_{11}} & \dots & \mathbf{u_{1i}} & \dots & \mathbf{u_{n1}} \\ \dots & & & & \\ \mathbf{u_{i1}} & & \mathbf{u_{ii}} & \dots & \mathbf{u_{in}} \\ \dots & & & & \dots \\ \mathbf{u_{n1}} & & \mathbf{u_{ni}} & \dots & \mathbf{u_{i}} \end{pmatrix}$$

Na výstupu jsou pak v diagonále matice **R** vlastní čísla původní matice **R** a ve sloupcích pomocné matice **U** jsou vlastní vektory původní matice **R**. Barevně vyznačená jsou v 1. matici vypočtená vlastní čísla, ve 2. matici 1. vlastní vektor odpovídající 1. vlastnímu číslu.

Stopa eliminované matice **R** je (**λ1 + λ2 + … + λn**). Významnost jednotlivých vlastních čísel určíme jako procentuální podíl jednotlivých vlastních čísel vůči stopě, tedy

$$\boldsymbol{\lambda i * 100 / (\lambda 1 + \lambda 2 + \dots + \lambda n)}$$

Pokud významnost vlastního čísla je vůči stopě malá, příslušné hlavní komponenty můžeme zanedbat a snížit tak počet nových atributů.

❑ Hlavní komponenty

Je-li spočítána matice **U** s vlastními vektory korelační matice **R**, můžeme vypočíst hledané skryté proměnné jako hlavní komponenty původních atributů. Transformace do hlavních komponent se provádí pomocí vlastních vektorů matice korelační:

$$\mathbf{Z} = \mathbf{X}\,\mathbf{U}$$

kde **X** je původní matice dat, **Z** je transformovaná matice dat, **U** je matice vlastních vektorů vypočtených z matice korelační **R** pro všechny dvojice atributů x_i, x_j matice dat **X**.

Po prvcích jde o transformační rovnici

$$z_{ij} = \sum_{k=1}^{n} (x_{ik} * u_{kj})$$

Výsledkem je nová tabulka dat, v níž jsou původní (případně závislé) atributy {A1, …, An} nahrazeny vzájemně nezávislými hlavními komponentami {H1, …, Hn}, kde každá komponenta je jistou lineární kombinací původních atributů. Podle významnosti vlastních čísel je možno z n komponent uvažovat jen několik prvních dostatečně významných a snížit tak počet nových atributů.

Problémem někdy může být pojmenování nových komponent.

Algoritmus výpočtu hlavních komponent

Dána datová matice **X** (= tabulka s numerickými - reálnými atributy).

1. výpočet korelační matice **R** (příp. kovarianční K)
2. výpočet vlastních čísel **λi** a vlastních vektorů $\mathbf{u_i}$ této matice **R (K)**
3. uspořádání **λi** sestupně a uložení odpovídajících vektorů $\mathbf{u_i}$ do sloupců matice **U**
4. transformace datové matice X do nové datové matice rovnicí **Z** = **X U**

Příklad 3.3.[14]

V souboru jsou údaje o 104 plavkyních testovaných ve 4 stylech a 2 délkách tratě. Úkolem zjistit, zda „plavecká schopnost" je souhrn více nezávislých vlastností nebo jediná.

Data: plav (50 kraul, 200 kraul, 50 prsa, …,50 znak, … 50 delfín, …)

Výsledek: *Hlavní komponenty*

z1= **0.88** x1+ **0.82** x2 + 0.27 x3 + 0.20 x4 + **0.77** x5 + **0.76** x6 + **0.70** x7+ **0.57**x8
z2= 0.19 x1+0.26 x2 + **0.78** x3 + **0.77** x4 + 0.21 x5 + 0.35 x6 + 0.40 x7+ 0.48x8

Zjednodušené vyjádření hlavních komponent pomocí zaokrouhlení spočítaných složek vlastních vektorů na celá čísla - (zvýrazněných) koeficientů s vyšší absolutní hodnotou na hodnotu 1 a ostatních na hodnotu 0:

z1= x1+ x2 + x5 + x6 + x7+x8
z2= x3 + x4

Nyní můžeme říct, že existují 2 hlavní faktory, ovlivňující plaveckou schopnost, případně je pojmenovat.

Trenér plavkyň pojmenoval první dvě hlavní komponenty:

z1 *...schopnost plavat převážně pomocí paží*
z2 *...schopnost plavat převážně pomocí nohou*

Závěr: *existují 2 typy plavkyň, jedentyp je nadaný pro styl prsa, druhý typ je nadaný pro ostatní styly.*

♦

❑ Využití výsledků hlavních komponent

Výsledky výpočtu hlavních komponent se mohou využít

1. seskupením atributů spojenou s definováním skrytých proměnných,
2. k redukci počtu původních proměnných a tak ke zjednodušení popisu objektů,
3. ke zprostředkovanému měření nepřímo měřitelných proměnných a jejich odhadu,
4. k transformaci původních proměnných do výhodnějšího tvaru, spojené s jejich ortogonalizací.

Příklad 3.4. **[11]**

Na Pedagogické fakultě v Ostravě v 80. (komunistických) letech hodnotili úroveň výchovně vzdělávacího procesu u 32 absolventů - učitelů chemie po 1 roce praxe. Každý učitel byl hodnocen svým patronem, vedením školy a inspektorem bodováním 0 - 4 bodů v následujících 28 ukazatelích:

1. Splnění povinností daných učebním plánem.
2. Vědecká úroveň přednášek.
3. Dodržení kontinuity přednášeného učiva a využití mezipředmětových vztahů.
4. Spojení teorie s praxí v probíraném učivu.
5. Výchovné využití učiva.
6. Tvůrčí schopnosti a využívání nových metod ve výuce.
7. Osvětlování a využívání dialektických souvislostí při upevňování a opakování látky.
8. Názornost ve výuce a používání didaktické techniky.
9. Funkční začlenění chemických pokusů do výuky.
10. Kontakt učitele s žákem a učební klima při výuce.
11. Podnícení zájmu žáků o chemii.
12. Aktivita žáků při vyučování.
13. Smysl pro pořádek a estetiku.
14. Osobnost učitele, jeho vystupování, znalosti, dovednosti, osobní kladné vlastnosti.
15. Veřejně politická angažovanost učitele.
16. Spolupráce učitele s PO SSM.
17. Spolupráce učitele s rodiči.
18. Sebevzdělávání a sebevýchova učitele.
19. Pedagogická dokumentace.
20. Spolupráce učitele s kolektivem školy.
21. Výchovné využití politického vzdělávání.
22. Diferencovaný přístup k žákům.
23. Agitační práce ve škole.
24. Všeobecná iniciativa a smysl pro pokrok.
25. Věcné znalosti.
26. Metodické znalosti.
27. Úroveň společenskopolitické praxe.
28. Zhodnocení fyzických podmínek pro výkon učitelského povolání.

Data není nutno předzpracovávat ani transformovat. Pro hlavní komponenty byly spočítány:

Stopa kovarianční matice = 19.37

Vlastní čísla : $\lambda = \{7.60, 2.34, 0.65, \ldots\}$

Procentuální významnost vlastních čísel: {39.2%, 12.1%, 3.3%, …}

Vlastní vektory :

u1 = (0.02, 0.13, 0.12, 0.10, 0.10, **0.28**, 0.11, 0.11, 0.17, **0.27**, 0.10, **0.25**, 0.15, 0.15, **0.29**, 0.02, -0.03, **0.26**, 0.06, 0.12, 0.12, 0.13, 0.21, **0.29**, 0.18, 0.15, 0.11, -0.02)
u2 = (-0.10, -0.12, -0.14, -0.17, -0.09, -0.12, -0.08, -0.14, -0.09, -0.08, -0.21, -0.14, -0.10, -0.17, **0.63, 0.46**, -0.05, 0.07, 0.05, 0.05, 0.09, 0.08, 0.17, -0.04, 0.01, -0.08, **0.24**, 0.19)
...

*První vlastní vektor má nejvyšší hodnoty u atributů číslo 6, 10, 12, 15 a 24. Jejich společný rys bychom mohli nazvat „míra **zápalu pro učitelství**“. Hodnota prvního vlastního čísla je 39% stopy, tedy první nejvýraznější faktor variability je dosti výrazný.*

*Druhý vlastní vektor má nejvyšší hodnoty s čísly 15, 16 a 27, jejich společný rys můžeme shrnout jako „míra **společenské aktivity**“. Hodnota vlastního čísla je 12 % stopy, tedy druhý faktor je výrazný mnohem méně a znamená větší vyrovnanost absolventů.*

Další vlastní čísla vyšla relativně velmi malá, proto nemá smysl uvažovat o dalších faktorech.

***Závěr**: existují 2 hlavní faktory výše pojmenované, rozlišující mezi sebou učitele; pokud by bylo možnohodnotit přímo jen tyto 2 vlastnosti, stačily by hodnotitelům 2 atributy, pokud není možno přímo najít pravidlo pro jejich hodnocení, stačilo by místo původních 28 otázek položit hodnotitelům jen otázky 6, 10, 12, 15, 16, 24 a 27 (tedy jen 7) a z nich zjednodušenými transformačními rovnicemi spočítat tyto 2 vlastnosti:*

z1 = 28.x6 + 27.x10 + 25.x12 + 29.x15 + 26.x18 + 29.x24
z2 = 63.x15 + 46.x16 + 24.x27

V obou případech dojde k výraznému zmenšení počtu atributů bez výrazného omezení informace o objektech. Menší počet nezávislých atributů v dalších metodách DM

♦

Příklad 3.5.

Poctivý génius, který lže a krade

Člověk bývá charakterizován množstvím vlastnost. Otázkou je, zda se všechny vlastnosti dají popsat menším množstvím „reprezentativních“ vlastností. Výzkum na toto téma prováděla skupina analytiků po světě. Z původních mnoha atributů, které byly sbírány o lidech:

Člověk (ustaranost, hněvivost, sebevědomí, impulsivnost, zranitelnost, vřelost, společenskost, aktivita, citlivost, důvěřivost, podrobivost, umírněnost, něžnost, poslušnost, cílevědomost, disciplinovanost, uvážlivost, ...)

⇓ *faktorová analýza našla 5 nezávislých komponent:*

Člověk (emoční stálost, otevřenost, extroverze, přátelskost, svědomitost)

emoční stálost = míra odolnosti vůči záporným pocitům
vysoce reaktivní ⇔ kliďas
otevřenost = míra zvědavosti na vnitřní a vnější svět
badatel ⇔ konzervativní „udržovatel“
extroverze = míra udržování aktivních vztahů k jiným lidem
extrovert ⇔ introvert, do sebe zahleděný
přátelskost = měřítko altruismu
altruista ⇔ egocentrista
svědomitost = míra sebekontroly vůle něčeho dosáhnout
soustředěnost na osobní cíle ⇔ pružný, požitkářský

⇓

průzkum mezi lidmi, názory na sebe, známé osobnosti, politiky

Výsledek:

- sebe a známé osobnosti (z kultury, filmu, sportu, ...) hodnotí dotazovaní ve všech 5 rozměrech
- politiky hodnotí pouze ve 2 rozměrech, a to ještě značně závislých:

  1. jak politik dovede přesvědčit okolí o tom, že je poctivý, pravdomluvný, odpovědný, spolehlivý, precizní, vytrvalý, umírněný, velkorysý a citově vyrovnaný

  2. jak politik dovede přesvědčit okolí o tom, že je aktivní, přiměřeně sebejistý, energický, tvořivý, chytrý, moderní, výkonný, vynalézavý a srdečný.

Bez ohledu na skutečné vlastnosti hodnotí jen herecké schopnosti. Politikovi stačí hrát roli + pomluvit protivníka (efekt spáče).

♦

Shrnutí pojmů 3.

Korelace dvou veličin.
Hlavní komponenty a jejich konstrukce.
Reprezentace dat pomocí lineárních kombinací původních veličin.
Využití výsledků hlavních komponent

Otázky 3.

1. Definujte korelaci dvou veličin – atributů.
2. Popište podstatu skrytých faktorů (hlavních komponent) pomocí známých veličin.
3. Popište metodu výpočtu hlavních komponent.
4. K čemu slouží výsledek výpočtu hlavních komponent?

Úlohy k řešení 3.

1. Jsou dána data jednoho ostravského fitcentra. V něm nabízejí jako službu svým zákazníkům tzv. „zjištění celkové kondice těla“. K výsledku je zapotřebí znát některé vlastnosti daných osob, proto si fitcentrum vytvořilo dotazník pro jejich zjištění, další data naměřili a vypočetli.

   U jednotlivých osob byly sledovány tyto vlastnosti:

   Základní údaje

   - id – identifikace osoby
   - věk – celé číslo udávající počet roků
   - pohlaví – muž/žena
   - klidová TF – tepová frekvence/minutu
   - krevní tlak – nízký/normální/vysoký

Tělesné proporce

- výška – v centimetrech
- váha – v kilogramech
- obvod krku – v centimetrech
- obvod bicepsu – v centimetrech
- obvod hrudi – v centimetrech
- obvod pasu – v centimetrech
- obvod boku – v centimetrech
- obvod hýždí – v centimetrech
- obvod stehna – v centimetrech
- obvod lýtka – v centimetrech

Aerobní zdatnost

- ergometr - watt/kilogram
- VO_2MAX - mililitr kyslíku/kilogram/minuta
- step test - tepová frekvence/minuta

Test síly

- sedy-lehy – počet sedů lehů za 1 minutu
- kliky – počet kliků lehů za 1 minutu
- benchPress1 – v kilogramech
- benchPress2 – v kilogramech
- legPress – v kilogramech

Antropologické měření

- PT – procento tuku v těle - v procentech
- A.T.H – hmotnost bez tuku – v kilogramech
- H.T – hmotnost tuku – v kilogramech
- BMI – Body Mass Index = index tělesné zdatnosti (výška [cm]/hmotnost [kg]) -
- WHR – Waist to Hip Ratio = Poměr pasu a boků (obvod pasu/obvod boků)

Pro tato data navrhněte použití metody hlavních komponent.

Semestrální projekt Analýza 3

Pro svá data poveďte analýzu použitelnosti metody hlavních komponent. Pokud je metoda použitelná, navrhněte její využití. Pokud není, zdůvodněte to.

4. ASOCIACE

Čas ke studiu: 6 hodin

Cíl Po prostudování této kapitoly budete

- vědět, co jsou asociace mezi podmnožinami atributů a jaké typy asociací rozeznáváme,
- umět pro každý typ asociací popsat a zdůvodnit jejich výpočet,
- pomocí popsaných metod vybrat vhodnou metodu, analyzovat a řešit praktické úlohy generování asociací různých typů.

Výklad

4.1. Asociace jako vztahy mezi atributy

Jak víme, zkoumané objekty jsou popsány svými atributy. Jednou velkou skupinou, nejčastěji využívanou při dolování znalostí, je hledání vztahů – asociací mezi podmnožinami atributů.

Představme si objekty v databázi jako „nosiče“ mnoha vlastností. Může se stát, že mezi vlastnostmi jsou jisté, ne vždy zřetelné vztahy. Pokud experta nebo analytika „napadne“, že by mezi atributy A a B mohl být vztah například typu „jestliže A=1, pak B=5“, může si tuto vlastní hypotézu v datech otestovat. Ovšem, pokud takový předpoklad neformuluje, také ho neotestuje. Přitom mohou existovat zajímavé vztahy, které nikdo nerozpozná, protože nikoho nenapadne se na ně ptát.

Prapůvodem metody hledání asociací je metoda GUHA [7] - původní česká metoda pražské skupiny autorů (P. Hájek, M. K. Chytil, T. Havránek) z roku 1964. Od té doby se rozšířila po celém světě.

Metoda používá prostředků matematické statistiky a matematické logiky. Automaticky a systematicky generuje a ověřuje hypotézy těchto typů:

1. A (asi, většinou) souvisí s B
2. A (asi, většinou) je příčinou B

Pojmy „souvisí“, „je příčinou“, budou upřesněny níže, veličinami A, B rozumíme podmnožiny atributů zkoumaných objektů (původní, filtrované, odvozené).

Později, zvláště se začátkem využívání v marketingu, se rozvinuly i varianty základních asociací. Princip hledání všech možných asociací však zůstává stejný - jejich automatické generování a testování, což to je podstata rozdílu proti testování v matematické statistice.

Asociacemi tedy nazýváme vztahy mezi podmnožinami atributů. Rozlišujeme asociace

- **klasické** - mezi dvěma podmnožinami atributů v relačních datech,
- **transakční** - v rámci rozsáhlé množiny atributů, zaznamenaných seznamem jejich výskytů,
- **agregované** - mezi podmnožinou atributů a jejich skupinovými charakteristikami.

4.2. Asociace základní

❑ Základní pojmy metody asociací

Metoda pro získávání asociačních pravidel z relačních dat je nejčastěji používanou metodou v DM. Vychází z uvedené metody GUHA. Hledá asociace – vztahy mezi podmnožinami atributů, formulované ve tvaru pravidel. K jejich popisu zavedeme několik nových pojmů.

Mějme nejprve binární datovou matici **XB** s objekty {O1, O2, …} a atributy {A, B,…, X, Y, …} ve sloupcích.

XB

	A	**B**	**…**	**X**	**Y**	**…**
O1	x11	x12		…	…	
	…	…				
Oi	xi1	xi2				
	…	…				
Om	xm1	xm2				

Jména atributů A, B, … nazveme jednoduchými binárními **formulemi**, obecně je označíme **Fi**.

Z binárních formulí je možno vytvářet **složené binární formule** pomocí logických spojek negace, konjunkce a disjunkce. Jsou-li Fi, Fj dvouhodnotové formule, pak také

$$\neg \mathbf{Fi} \qquad \mathbf{Fi} \wedge \mathbf{Fj} \qquad \mathbf{Fi} \vee \mathbf{Fj}$$

jsou dvouhodnotové formule.

Pro asociace mají význam **elementární konjunkce**, tj. formule tvaru

$$\pm \mathbf{F1} \wedge \pm \mathbf{F2} \wedge \ldots \wedge \pm \mathbf{Fk}$$

a **elementární disjunkce** tvaru

$$\pm \mathbf{F1} \vee \pm \mathbf{F2} \vee \ldots \vee \pm \mathbf{Fk}$$

kde symbol ± znamená, že před Fi je negace nebo nic; Fi jsou navzájem různé veličiny.

Příklad 4.1.

Pro zdrojovou datovou matici Pacient (jméno, věk, váha, tlak, …, kouří, pije_alkohol, užívá_drogy, vegetarián, …, vysoký_tlak, rakovina_plic, cukrovka, infarkt, …) *jsou sledovanými objekty pacienti s některou z evidovaných nemocí.*

Jednoduché formule jsou například: kouří *[A/N],* vegetarián *[A/N],*

složené formule jsou například: pije_alkohol ∧ ¬kouří ∨ vegetarián.

Elementární konjunkcí je například: ***cukrovka ∧ ¬infarkt ∧ vysoký_tlak***

Elementární disjunkcí je například: ***kouří ∨ pije_alkohol ∨ užívá drogy***

Vztahy, které mohou lékaře zajímat, jsou například vztahy mezi atributy charakterizujícími styl života na jedné straně a atributy popisujícími diagnózu na straně druhé.

♦

❑ Čtyřpolní tabulka pro binární data

Pro binární (později i kategoriální) data dávají užitečnou informaci **frekvence** = četnosti výskytů. Zapisujeme je do tvaru frekvenční tabulky.

Pro data **XB** a z nich vytvořené binární formule F1 a F2 má frekvenční (čtyřpolní) tabulka tvar

platí F1 \ F2	ano	ne	∑
ano	**a**	**b**	**r**
ne	**c**	**d**	**s**
∑	**k**	**l**	**m**

kde **a** je počet případů, kdy F1=F2=1 (počet kladných shod),

b počet případů F1=1, F2=0,

c počet případů F1=0, F2=1 (b+c je počet neshod),

d počet případů F1=F2=0 (počet záporných shod).

Příklad 4.2.

Mějme jednoduchá data z tabulky X s binárními atributy. Pak čtyřpolní tabulka pro atributy A1 a A2:

X

A1	A2	…	An
1	0	…	…
1	1		
0	1		
0	1		
0	0		
1	0		
1	0		
0	0		
1	1		

A1\A2	1	0	∑
1	**2**	**3**	**5**
0	**2**	**2**	**4**
∑	**4**	**5**	**9**

♦

❑ Kvantifikátory, sentence a hypotézy

Intuitivně vytušíme, že budou-li počty shod výrazně vyšší než počty neshod, bude mezi formulemi F1 a F2 nějaký vztah, v opačném případě budou obě formule vzájemně nezávislé. Abychom takové vztahy mohli kvantifikovat, zavedeme jejich numerické charakteristiky, nazývané zobecněné **kvantifikátory.** Různé typy vztahů jsou definovány různými kvantifikátory. Všechny kvantifikátory jsou vypočteny z hodnot čtyřpolní tabulky.

Uvedeme si dva nejpoužívanější kvantifikátory, odpovídající v úvodu kapitoly uvedeným vztahům F1 souvisí s F2, F1 je příčinou F2.

I. Kvantifikátor asociační pro kvantifikaci vztahu „souvisí s“.

II. Kvantifikátor implikační pro kvantifikaci vztahu „je příčinou“.

Kvantifikátory obecně označíme **q**, nabývají hodnot 0 (vztah neplatí) nebo 1 (vztah platí).

Zavedeme si ještě další pojmy, užívané v souvislosti s asociačními pravidly. **Sentencí** nazveme testované pravidlo asociační nebo implikační. Formuli na levé straně sentence nazýváme **antecedentem**, formuli na pravé straně **sukcedentem.**

Sentence **je pravdivá** na datech **XB**, jestliže po aplikaci zvoleného kvantifikátoru na datech dostaneme hodnotu 1. Sentence platné na datech nazveme **hypotézami.**

Pro danou matici dat **XB** je dále dána množina sentencí, které jsou považovány z nějakých důvodů za **relevantní otázky**. Množina relevantních otázek je popsána pravidly, jak je generovat: zadáním formulí antecedentu a sukcedentu, zadáním typu kvantifikátoru a zadáním některých dalších parametrů pro jejich generování. Zajímá nás, které relevantní otázky jsou pravdivé v datech **XB**.

Příklad 4.3.

Pro data Pacient *použijeme binární atributy*

Pacient (jméno, věk, váha, tlak, …, kouří, pije_alkohol, užívá_drogy, vegetarián, …, vysoký_tlak, rakovina_plic, cukrovka, infarkt, …)

mohou být příkladem množiny relevantních otázek sentence

s antecedentem (pije_alkohol ∧ kouří ∧ užívá drogy ∧ ¬vegetarián),

se sukcedentem (cukrovka ∨ infarkt ∨ vysoký_tlak)

a s vhodným implikačním kvantifikátorem. Relevantní otázky tak formulují vztah mezi stylem života a diagnózou. Formulováno jako pravidlo:

Jestliže (pije_alkohol ∧ kouří ∧ užívá drogy ∧ ¬vegetarián)

pak (cukrovka ∨ infarkt ∨ vysoký_tlak)

♦

❑ Kvantifikátory asociační pro binární veličiny

odpovídají intuitivní představě souvislosti dvou množin atributů: kvantifikátor q je asociační, jestliže sentence **F1 q F2** říká, že F1 souvisí s F2 (že shody jejich hodnot převažují nad neshodami).

Definice 4.1.

Kvantifikátor **q** je asociační, jestliže pro každé a, b, c, d takové, že q(a,b,c,d) = 1 a každé a'≥ a, b'≤b , c' ≤ c, d' ≥ d je q (a',b',c',d') = 1.

Asociační kvantifikátory jsou symetrické, tj. q(a, b, c, d) = q(a, c, b, d).

Jednotlivé asociační kvantifikátory jsou definovány takto:

1. prosté vychýlení $\sim\delta$ pro libovolné $\delta \geq 0$

$$\sim\delta\,(a,b,c,d) = 1 \quad \text{je-li} \quad ad > e^{\delta}\,.\,bc \qquad \dots \quad ad / bc > e^{\delta}$$

2. Fisherův kvantifikátor $\sim\alpha 1$ pro libovolné $\alpha \in (0, 0.5>$

$$\sim\alpha 1\,(a,b,c,d) = 1 \quad \text{je-li} \quad ad > bc \quad \text{a} \quad \sum_{i=a}^{\min(r,k)} \frac{\binom{k}{i}\binom{m-k}{r-i}}{\binom{m}{r}} \leq \alpha$$

Tento kvantifikátor je založen na testu hypotézy o nezávislosti testovaných veličin proti alternativě o jejich kladné závislosti; jde o test na hladině významnosti α. Hodnota 1 indikuje přijetí alternativní hypotézy.

Je vhodný pro malá $\mathbf{m \in \langle 20, 40 \rangle}$ nebo pro četnosti $\mathbf{a, b, c, d < 5}$.

Protože rozsah dat pro DM bývá obvykle výrazně větší, než 40 objektů, tento kvantifikátor se příliš často nepoužívá.

3. χ^2 **- kvantifikátor** $\sim\alpha^2$ pro libovolné $\alpha \in (0, 0.5\rangle$

$$\sim\alpha^2 \ (a,b,c,d) = 1 \quad \text{je-li} \quad ad > bc \quad \text{a} \quad \frac{(ad-bc)}{rkls} m \geq \chi^2_\alpha$$

kde χ_α^2 je $(1 - \alpha)$ - kvantil χ^2 rozložení s jedním stupněm volnosti; je to test asymptotický. Statistický význam má stejný, jako Fisherův test.

Je vhodný pro $\mathbf{m > 40}$ nebo pro $\mathbf{m \in \langle 20, 40 \rangle}$ a četnosti $\mathbf{a, b, c, d > 5}$.

Charakteristiky pro souvislosti (též asociace v užším slova smyslu jako "skoroekvivalence") odpovídají intuitivní představě souvislosti: kvantifikátor q je asociační, jestliže sentence říká, že antecedent souvisí se sukcedentem, že počet shod převládá nad počtem neshod.

Pokud ve čtyřpolní tabulce platí c=b=0, pak jde o logickou ekvivalenci.

Příklad 4.4.

Pro data Pacient (jméno, věk, váha, tlak, …, kouří, pije_alkohol, užívá_drogy, vegetarián, …, vysoký_tlak, rakovina_plic, cukrovka, infarkt, …)

s definovanými relevantními otázkami formou sentencí

pije_alkohol $\wedge$ kouří $\wedge$ užívá drogy $\wedge \neg$vegetarián $\sim\alpha^2$ cukrovka $\vee \neg$infarkt $\vee$ vysoký_tlak

vyjdou hodnoty ve čtyřpolní tabulce

platí F1 \ F2	ano	ne	∑
ano	122	14	136
ne	38	226	264
∑	160	240	400

Prosté vychýlení*: protože* a.d / b.c = 51,8 > e^δ *pro* $\delta = 1$, *platí* $\mathbf{\sim\delta(a,b,c,d) = 1}$ **...** *tedy je prokázáno prosté vychýlení antecedentu a sukcedentu,*

nebo

Fischerův test*: ze čtyřpolní tabulky je* (ad-bc)/(klrs)*m = 0,007845. *Tabulková hodnota* χ^2 *s jedním stupněm volnosti a s hladinou významnosti* $\alpha = 0{,}05$ *je* $\chi^2 = 6.6$ … *závěr:* $\mathbf{\sim\alpha^2(a,b,c,d)=0}$, tedy *Fisherův test na hladině významnosti* $\alpha = 0{,}05$ *nepotvrdil alternativní hypotézu o závislosti antecedentu a sukcedentu.*

Jak vidíme,použití různých kvantifikátorů může mít různé výsledky; záleží také na volbě parametrů δ *nebo* α**.**

♦

Asociační kvantifikátory se používají méně často, než následující implikační kvantifikátory, protože symetrický vztah atributů je v realitě méně častý.

❑ Kvantifikátor implikační pro binární veličiny

Je-li q kvantifikátor implikační, pak formule **F1 q F2** říká, že skoro všechny objekty splňující F1 splňují i F2.

Definice 4.2.

Kvantifikátor q je implikační, jestliže pro každé a, b, c, d takové, že q (a,b,c,d) = 1 a pro každé a' ≥ a, b' ≤ b a libovolné c', d' platí

$$q(a', b', c', d') = 1.$$

Implikační kvantifikátor není symetrický a je definován takto:

Fundovaná implikace (FI) $\Rightarrow_{Pmin,Smin}$ (pro Smin ∈ (0,1> a Pmin > 0)

$$\Rightarrow_{\mathbf{Pmin,Smin}}(\mathbf{a, b, c, d}) = \mathbf{1} \quad \text{je-li } a \geq Pmin \text{ a } a \geq Smin\ (a+b)$$

kde Smin ... minimální spolehlivost vztahu definována vhodným kvantifikátorem,

Pmin ... minimální podpora, minimální počet případů, pro něž je nalezená hypotéza platná.

Pro S = 1 a P = 0 dostáváme obvyklou logickou implikaci.

FI můžeme formulovat také ve tvaru pravidla

Platí-li **F1,** pak platí **F2** se spolehlivostí **S** a podporou **P**

kde P = a, S = a / (a+b).

Kvantifikátor **FI** je používán v DM nejčastěji. Odpovídá intuitivní představě vztahu příčina – následek a jeho výpočet je velmi jednoduchý.

❑ Asociace základní nad kategoriálními daty

Pro testování spolehlivosti asociace se používají statistické charakteristiky, které jsou definovány pro binární data a vypočtené ze čtyřpolní tabulky shod a neshod antecedentů a sukcedentů.

Skutečná data obvykle nejsou jen binární. Z kapitoly o předzpracování dat víme, že je možno data kteréhokoliv datového typu převést na binární. Pokud máme data alespoň kategoriální, pak je obvykle pro asociace "binarizují" přímo dolovací algoritmy. Například tak, že testují postupně hodnoty atributu A=a1, A=a2, …, A=ak.

Zobecníme-li definici asociací, pak asociacemi nyní nazýváme vztahy mezi dvěma podmnožinami kategoriálních atributů (antecedentem a sukcedentem).

Je to tedy jeden z typů tvrzení

Jestliže $A=a \wedge B=b \wedge \ldots$ **pak** $X=x \wedge Y=y \wedge \ldots$ **se spol** S **a podp** P (1)

Hodnoty $A=a \wedge B=b \wedge \ldots$ **souvisí s** $X=x \wedge Y=y \wedge \ldots$ **se spol** S **a podp** P (2)

kde $\{A,B,\ldots\}$ jsou atributy antecedentu,

$\{X,Y,\ldots\}$ jsou atributy sukcedentu,

$a, b, \ldots, x, y, \ldots$ jsou hodnoty domén příslušných atributů,

S ... spolehlivost vztahu definovaná vhodnými kvantifikátory,

P ... podpora, počet případů, pro něž je nalezená hypotéza platná.

Výsledkem jsou takové vygenerované hypotézy, pro něž platí

S ≥ **Smin = minconf** (minconf je zadaná minimální **spolehlivost, confidence**)

P ≥ **Pmin = minsupp** (minsupp je zadaná minimální **podpora, support**)

Příklad 4.5.

Mějme jednoduchou datovou tabulku **Z** *s atributy* A, B, … X, Y, …, *relevantní otázky jsou zadány sentencí typu fundované implikace*

Je-li $A=a \wedge B=b$ …, **pak** $X=x$ … **se spol** S **a podp** P

jedním z testovaných případů této sentence je

Je-li $A=1 \wedge B=2$, ***pak*** $X=3$ ***se spol*** S ***a podp*** P

Z

A	B	…	X	Y	…
1	2		3	4	
3	3		3	1	
3	4		2	1	
2	3		1	1	
1	2		3	2	
2	4		2	2	
1	3		3	2	
1	2		4	1	
2	4		1	2	
1	2		3	1	

⇒

ante	sukce
1	1
0	1
0	0
0	0
1	1
0	0
0	1
1	0
0	0
1	1

ant\suk	1	0	∑
1	3	1	4
0	2	4	6
∑	5	5	10

tato sentence se testuje pro každou nastavenou množinu hodnot atributů A, B, ... stejně, jako pro data binární; pomocná tabulka s vyhodnocením antecedentu a sukcedentu je zobrazena jen pro lepší pochopení. Její hodnoty se načítají přímo do čtyřpolní tabulky.

Pro zobrazené hodnoty $A=1 \wedge B=2$, $X=3$ *vychází*

FI = a/(a+b) = 3/4 = 0,75, *tedy spolehlivost* S = 75%, *podpora* P = a = 3.

♦

❑ Nastavení minimální podpory a spolehlivosti

Obecně se doporučuje začít s nastavením pro minimální podporu asi na 5% počtu záznamů v datech.

Pro hodnotu minimální spolehlivosti se doporučuje toto počáteční nastavení:

a) 90% pro přesná data, kde se neprojevuje nějaké subjektivní hodnocení, tedy např. data z lékařských měření, sportovní výsledky apod.

b) 70% pro data, kde se projevuje nějaké subjektivní hodnocení - typicky to jsou nějaká data získaná z dotazníků s otázkami typu „Jaký je Váš názor na to či ono?“ apod.

Zvyšováním minimální podpory klesá počet nalezených pravidel a naopak, zvyšováním minimální spolehlivosti také klesá počet nalezených pravidel a naopak. Těchto zákonitostí můžeme využít k získání souboru pravidel, který není ani příliš rozsáhlý, ani příliš malý.

Poznámka 1: Nemá smysl hledat pravidla se spolehlivostí 50%. Takové pravidlo říká toto: v případě, že platí podmínka z antecedentu, podmínka v sukcecedentu bude splněna s pravděpodobností stejnou, jakou je pravděpodobnost padnutí panny při hodu korunou. Rozhodování na základě takového pravidla tedy není příliš sofistikované.

Poznámka 2: Není vhodné nastavovat parametr minimální podpory příliš vysoko. Lepší je nejdříve snižovat počet nalezených pravidel zvyšováním minimální spolehlivosti a teprve až když je spolehlivost někde mezi 90% a 95% a soubor pravidel je stale příliš velký, tak potom lze počet pravidel snižovat zvyšováním minimální podpory. Zvyšováním minimální podpory se totiž můžeme připravit o závislosti, které se v datech nemají moc velkou podporu, přesto však mohou být zajímavé.

❑ Metody ASSOC a IMPL

Metodu pro hledání asociací v užším slova smyslu nazveme **ASSOC**, metodu pro hledání implikací nazveme **IMPL.**

Obě metody používají stejný algoritmus pro generování všech možných sentencí daného typu, oba kvantifikátory se vypočítávají ze stejné čtyřpolní tabulky, proto se mohou oba typy výsledků hledat současně.

Metoda ASSOC generuje a testuje zajímavé sentence tvaru

$$\mathbf{KONJ1} \Leftrightarrow_{\alpha 2} \mathbf{KONJ2}$$

Metoda IMPL generuje a testuje zajímavé sentence tvaru

$$\mathbf{KONJ} \Rightarrow_{\mathbf{Pmin,Smin}} \mathbf{DISJ}$$

kde KONJ, KONJ1, KONJ2 jsou elementární konjunkce,

DISJ je elementární disjunkce

Řešitel zadává relevantní otázky tím, že specifikuje

- použitý kvantifikátor a jeho parametry (α nebo Pmin, Smin);
- povolený tvar antecedentu – jeho atributy a případně maximální délku, tj. maximální počet antecedentových atributů;
- povolený tvar sukcedentu – jeho atributy a případně maximální délku, tj. maximální počet sukcedentových atributů;

Poznámka:

Některé SW systémy s DM metodami generují jen jednoduchý sukcedent, tedy ne sukcedent ve tvaru KONJ nebo DISJ. Některé SW systémy zase nerozlišují antecedenty a sukcedenty, používají všechny atributy na obou stranách sentencí; pak si musí řešitel vyhledat smysluplné hypotézy "ručně" sám vhodnou filtrací výsledků.

❑ Algoritmy hledání asociací

Rozdíl mezi klasickými statistickými testy, prováděnými nad daty, a mezi dolováním asociací z dat je v automatizaci tohoto procesu. Dolovací algoritmy automaticky generují a testují "všechny možné" hypotézy a generují na výstupu ty, které projdou příslušným testem (s vhodným testovacím kritériem a zadaným hodnotami *minconf, minsupp*).

Problém všech algoritmů hledajících asociace je v testování "všech možných" sentencí, uvážíme-li počet teoretických případů, které je nutno otestovat, a jim odpovídající časovou složitost úlohy. Proto hlavním problémem celé metody je nalezení rychlých algoritmů, které by sice nalezly všechny požadované hypotézy, ale omezily by počet testů.

Algoritmy pro testování a generování hypotéz můžeme dělit na

- *triviální* - generují a testují všechny možné sentence jako kombinace hodnot atributů, délek ante a sukce, kombinací dvojic ante a sukce (odtud též kombinační analýza, pro větší data nepoužitelná, s exponenciální časovou složitostí),
- *uspořádané generování pravidel* – například vhodně uspořádané postupné prodlužování délky sukce a ante snižuje počet kladných shod a, pokud a < *minsupp*, negenerují se další sentence s větší délkou –cedentu,
- *vzorkováním* - rozsáhlá data zpracují po částech (vzorcích) a hledají kandidáty pro hypotézy, pak testují přes celá data jen kandidáty.

❑ Problém redundance výsledků

Při mechanickém generování asociací může nastat následující situace:

Platí-li $A=a \Rightarrow X=x$ se spolehlivostí $S = s$ (1)

a také $A=a \wedge B=b \Rightarrow X=x$ se spolehlivostí $S = s'$ (2)

Pak pro $s' = s$ říkáme, že (2) je redundandní (B=b je redundandní, nepřináší novou informaci)

$s' > s$ říkáme, že $B=b$ rozšiřuje sentenci (1)

$s' < s$ říkáme, že $B=b$ zužuje sentenci (1)

Za zajímavé obvykle považujeme rozšíření již nalezených hypotéz, naopak zúžení obvykle přiřadíme k redundancím.

Možnost rozšíření definice redundance: za redundandní považujeme i případy s' < s+C pro dané C.

Příklad 4.6.

Data o asistované reprodukci (AR) obsahují léčebné cykly AR, každý je popsán asi 250 atributy o pacientce, anamnéze, léčbě, dále časově dělené do několika etap, o dílčích výsledcích etap cyklu i výsledku celé léčby.

Jeden z vygenerovaných výsledků (použita FI), metoda AH = asistovaný hashing.

Je-li věk = 20-25 ∧ metoda AR = AH,

pak počet oplodněných oocytů > 0

s podp P = 112 **a spol** S = 76%

***Interpretace**: V datech existuje 112 případů pacientek věku 20-25 let, u kterých byla provedena metoda AH a z nich bylo 76% oocytů oplodněno.*

***Vyhodnocení:** Procento oplodněných vajíček v celém souboru AR je asi 50%. Na základě těchto výsledků můžeme formulovat **hypotézu**: U mladších pacientek věku 20-25 let metoda AH pozitivně ovlivňuje oplození vajíček.*

♦

❑ Využití metod ASSOC a IMPL

Výsledky obou metod můžeme použít pro následující analýzy:

1. **analýza příčin** ... použijeme IMPL, setřídíme výsledky dle SUKCE
2. **analýza následků** ... použijeme IMPL, setřídíme výsledky dle ANTE
3. **analýza asociací** ... použijeme ASSOC, setřídíme výsledky podle potřeby
4. **konkrétní dotaz** ... použijeme metodu dle dotazu; při konkrétním dotazu máme možnosti:

   buď dotaz → analýza na míru → výsledek

   nebo provedení komplexní.analýzy → dotaz → výběr výsledku

5. **tvorba báze pravidel** ... použijeme IMPL, setřídíme výsledky, výsledná pravidla chápeme jako pravidla pro použití u dalších případů z reality

Příklad 4.7.

Matky, které daly své dítě k adopci si to někdy později rozmyslí. Úkolem je najít pravidla rozpoznávající, které atributy (příčiny) napovídají, zda si matka své rozhodnutí rozmyslí nebo ne.

Data jsou od 104 matek se 23 atributy

Matka (m-věk, m-povolání, m-stav, m-národ, o-stav, o-vztah, o-povol, o-národ, počet-těhot,

počet-potrat, ... , kojení, ... , rozmyslela)

Řešení: Zvolena metoda IMPL, kvantifikátor FI, délka antecedentu 3

Výsledek – část úplného řešení:

Je-li	**pak**	**Spolehlivost**	**Podpora**
m-věk < 30	rozmyslela	66	62
m_věk ∈ <30, 40>	nerozmyslela	100	28
m_věk > 40	nerozmyslela	91	14
m_povol = dělnice, pomocná	nerozmyslela	100	19
o_povol = není ve vězení	nerozmyslela	90	86
o_vztah = ženatý jinde	nerozmyslela	100	67
poč_těhot = 3 ∧ poč_potrat = 0	nerozmyslela	100	22
útěk = ano	nerozmyslela	100	17
steril = ano	nerozmyslela	100	6
m_povol = prostituce	nerozmyslela	100	15
ústavní_výchova = ano	nerozmyslela	100	21
podvod s OP = ano	nerozmyslela	100	5
…			

Výsledek formulovaný slovně, uspořádaný podle síly pravidla:

Jestliže platí

m-věk>= 30 let nebo
m-povolání = dělnice,pom.síla,THP nebo
o-vztah=ženatý s jinou, rozvedený nebo
poč-těhot>=3 ^ poč-potratů=0 nebo
prostituce = ano

pak rozmyslela = ne
s podporou P = … **a spolehlivostí = 100%**

Jestliže platí
m-věk <= 30 let ^ o-věk <= 30 let
pak rozmyslela = ne
s podporou P = ... **a spolehlivostí** = **66%**
Jestliže platí
m-věk <= 18 let
pak rozmyslela = ne
s podporou P = ... **a spolehlivostí** = **57%**
...

♦

Příklad 4.8.

Zadání: Japonská banka sleduje data o 125 lidech, žádajících banku o úvěr. Úkolem je z dosavadních „ručně" prováděných rozhodnutí formulovat pravidla, podle kterých se bude v budoucnu rozhodovat automaticky. Evidence má atributy:

A1. dostal úvěr [A/N]
A2. je nezaměstnaný [A/N]
A3. čím ručí [1 = PC, 2 = car, 3 = stereo, 4 = bike]
A4. pohlaví
A5. svobodný [A/N]
A6. problematický region [A/N]
A7. věk [kateg]
A8. hotovost u banky
A9. měsíční příjem
A10. počet měsíců splátky
A11. počet roků pracujících v současné firmě

Řešení: Metoda IMPL,

parametry ANTE = {A2 .. A11}, SUKCE = A1

kvantifikátor = FI
min. spolehlivost S = 90%, min. podpora P = 1
délka ante = 3

Výsledky:

Je-li věk = (65, 81> **pak** úvěr nedostal s S =100%

Je-li hotovost = (40,50 > a současně měs.příjem = (6, 10> **pak** úvěr nedostal s S =100%

Je-li nezaměst = Ano a současně měs.příjem = (6, 10> **pak** úvěr nedostal s S =100% ...

Je-li roků u firmy = >3 a současně pohl = muž a současně počet splátek = (6, 20>

pak úvěr dostal s S = 96.7%

♦

❑ Dedukční pravidla

Ze sentencí pravdivých v datech je možno jistými úpravami odvozovat další pravdivé sentence. Pravidla pro takové odvozování nazýváme dedukčními pravidly. Pomocí dedukčních pravidel je možno odvozovat další pravidla a není nutné je vyhledávat v datech.

Platí:

F1 a F2 jsou ekvivalentní v datech, platí-li Oi(F1)=Oi (F2) , i=1,..., m

1. Pravidlo záměny ekvivalentních formulí.

V každé sentenci S(F) obsahující formuli F a pravdivé v datech X lze nahradit formuli F jinou formulí F', která je v datech ekvivalentní formuli F. Vzniklá sentence je opět pravdivá v datech X.

Pro elementární konjunkce **K** a elementární disjunkce **D** platí zákony logiky: komutativní, negace-negace, De Morganovy ⇒ pro danou formuli tvaru **K** nebo **D** existuje řada **logicky ekvivalentních** formulí.

Příklad 4.9.

v datech X je pravdivá sentence

kouření $\Rightarrow_{0.8}$ infarkt $\vee$ rakovina_plic

pak jsou v datech pravdivé i sentence

kouření $\Rightarrow_{0.8}$ rakovina_plic $\vee$ infarkt

kouření $\Rightarrow_{0.8}$ $\neg$ $\neg$ (infarkt $\vee$ rakovina_plic)

kouření $\Rightarrow_{0.8}$ $\neg$ ($\neg$infarkt $\wedge$ $\neg$ rakovina_plic)

♦

Další ekvivalence mohou vyplynout z dat: 2 formule jsou **ekvivalentní v datech X**, mají-li pro všechny objekty shodné hodnoty: Oi(F) = Oj(F) pro všechna i, j.

Příklad 4.10.

v datech X platí

maj_auta je ekvivalentní s **maj_auta $\wedge$ muž**

pak ze sentence pravdivé v datech X (ne v realitě)

maj_auta $\Rightarrow_{0.9}$ má_garáž

pak i následující tvrzení jsou pravdivá v datech X (pravidlo 1)

maj_auta $\wedge$ muž $\Rightarrow_{0.9}$ má_garáž

muž $\wedge$ maj_auta $\Rightarrow_{0.9}$ má_garáž

♦

Praktické využití: stačí testovat elementární konjunkce a disjunkce jednoho pořadí

2. Pravidlo úprav elementární implikace.

Elementární implikací je sentence tvaru K Þ* D kde K je elementární konjunkce, D je elementární disjunkce a K, D nemají žádný společný atribut. Platí: v elementární implikaci K Þ* D pravdivé v datech X můžeme

(1) převést člen z antecedentu do sukcedentu za současné změny znamení (formuli A nahradit negací);

(2) přidat do sukcedentu nové členy.

Vzniklá sentence je opět pravdivá v datech.

Příklad 4.11.

v datech X je pravdivá sentence

$$\textbf{otylost} \wedge \textbf{muž} \;\text{Þ}_{0.8}\; \textbf{infarkt}$$

odtud plyne i pravdivost v datech pro sentence

$$\textbf{otylost} \wedge \textbf{muž} \Rightarrow_{0.8} \textbf{infarkt} \vee \textbf{ang.pect} \vee \textbf{berc.vred}$$

$$\textbf{otylost} \Rightarrow_{0.8} \neg\, \textbf{muž} \vee \textbf{infarkt} \vee \textbf{ang.pect}$$

$$\textbf{muž} \Rightarrow_{0.8} \neg\textbf{otylost} \vee \textbf{infarkt}$$

♦

Praktické využití:

- z pravdivosti jedné sentence můžeme rychle logicky odvodit (bez ověřování v datech) množství dalších sentencí pravdivých v datech
- při vhodném rozdělení atributů na ante – sukce je možno optimalizovat délku ante - sukce pro konkrétní algoritmus, skutečné rozdělení na ante – sukce pak použitím pravidla 2 upravit

3. Pravidlo symetrie.

Je-li ~ * symetrický kvantifikátor a je-li sentence F1 ~ * F2 pravdivá v datech X, pak i sentence F2 ~ * F1 je pravdivá v datech.

Příklad 4.12.

v datech X je pravdivá sentence

$$\textbf{kouření} \wedge \textbf{víc\_30\_let} \Leftrightarrow_{0.8} \textbf{rakovina\_plic}$$

pak jsou v datech pravdivé i sentence

$$\textbf{rakovina\_plic} \Leftrightarrow_{0.8} \textbf{kouření} \wedge \textbf{víc\_30\_let}$$

♦

Praktické využití:

při testování symetrických asociací stačí jeden test

3. Pravidlo konzervativního zlepšování.

Řekneme, že atribut A (nebo ¬A) konzervativně zlepšuje elementární konjunkci K v datech X, jestliže A se nevyskytuje v K a formule K ∧ A (nebo K ∧ ¬A) je ekvivalentní formuli K v X. V sentenci pravdivé v datech a obsahující elementární konjunkci K lze ke K přidat do konjunkce libovolný počet atributů a negovaných atributů, které K konzervativně zlepšují. Výsledná sentence je opět pravdivá v datech. Jde o speciální případ pravidla 1.

Příklad 4.13.

platí-li

$$\textbf{A1} \wedge \neg\textbf{A3} \wedge \textbf{A5} \Rightarrow \textbf{A4}$$

a zlepšují-li A5, ¬A7, A9 konzervativně antecedent, platí i

$$\textbf{A1} \wedge \neg\textbf{A3} \wedge \textbf{A5} \Rightarrow \textbf{A4}$$

$$\textbf{A1} \wedge \neg\textbf{A3} \wedge \textbf{A7} \Rightarrow \textbf{A4}$$

$$\textbf{. . .}$$

A1 ∧ ¬A3 ∧ ¬A7 ∧ A9 ⇒ A4
A1 ∧ ¬A3 ∧ A5 ∧ ¬A7 ∧ A9 ⇒ A4

♦

Praktické využití: jako u pravidla 1

❑ Asociace s neúplnou informací

Dána datová matice **X** binární s chybějícími údaji, tedy hodnotami {0, x, 1}, kde x je neznámá hodnota.

Dvouhodnotovým doplněním X je jsou taková data **X'**, že se x nahradí 0 nebo 1. Zřejmě jen jediné doplnění je správné (odpovídá realitě), ale to neznáme. Proto bereme v úvahu všechna možná doplnění a podle všech jejich výsledků můžeme dělat závěry.

Platí **princip zabezpečení**: není.li výsledek jistý, platí hodnota x.

Jestliže A1, . . . , An jsou atributy antecedentu a sukcedentu, pak

K = O(A1 ∧ A2 ∧ . . . ∧ An) =	1	když	A1 = A2 = . . . = An = 1
	0		některé Ai = 0, ostatní Aj = 1
	x		některé Ai = x
D = O(A1 ∨ A2 ∨ . . . ∨ An) =	1	když	některé Ai = 1
	0		A1 = A2 = . . . = An = 0
	x		některé Ai = x

Chápeme-li 0 < x < 1, pak

O (A1 ∧ A2 ∧ . . . ∧ An) = min (O(A1), . . . , O(An))

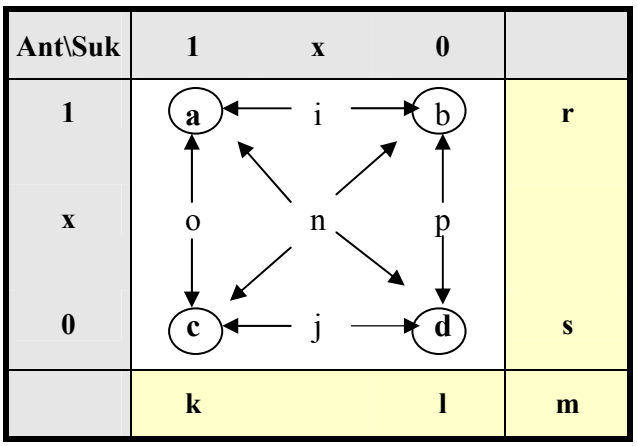
O (A1 ∨ A2 ∨ . . . ∨ An) = max (O(A1), . . . , O(An))

Princip zabezpečení říká, že elementární **K** nebo **D** má pro libovolný objekt hodnotu x právě tehdy, když existuje doplnění, že K(D) = 0 a jiné, že K(D) = 1.

Trojhodnotová frekvenční tabulka se převádí na dvojhodnotovou tak, že se

i se dílem přičte k **a**, dílem k **b**

o -"- **a** -"- **c** atd.



Příklad 4.14. **[7]**

Data o 15 pacientech Pacient(..., kouření, otylost, vysoký_tlak) *s hodnotami NULL.*

Označíme konjunkci kouření ∧ otylost = riziko

antecedent: riziko, *sukcedent*: tlak

testujeme implikaci ***riziko ⇒ tlak***

Trojhodnotová tabulka:

riziko \ tlak	**1**	**x**	**0**
1	**3**	**0**	**0**
x	**2**	**0**	**0**
0	**1**	**0**	**9**

pro tuto tabulku existují 3 možná doplnění:

(1)

5	0
1	9

(2)

4	0
2	9

(3)

3	0
3	9

*jsou pravdivá všechna (a/(a+b)=1) ⇒ sentence je pravdivá v datech (=**1**)*

♦

Příklad 4.15.

Data Pacient(..., kouření, otylost, vysoký_tlak) *s NULL.*

antecedent: kouření ∧ otylost = riziko

testujeme implikaci **¬riziko ⇒ ¬tlak**

Trojhodnotová tabulka:

¬riziko \ ¬tlak	**1**	**x**	**0**
1	**9**	**0**	**1**
x	**0**	**0**	**2**
0	**0**	**0**	**3**

pro tuto tabulku existují 3 možná doplnění:

(1)

9	1
0	5

(2)

9	2
0	4

(3)

9	3
0	3

je pravdivé 1. doplnění, nepravdivé 2. a 3. ⇒ sentence má v datech hodnotu **x.**

♦

Závěr: Data s větším počtem objektů a větším počtem neúplných hodnot v antecedentu a sukcedentu (tj. s velkými četnostmi i, o, n, p, j) jsou prakticky nezpracovatelná. Pro každou sentenci by se muselo testovat velké množství možných doplnění a výpočet by se velmi prodražil.

4.3. Asociace transakční

Jiný častý tvar zdrojových dat pro asociace je tzv. nákupní košík. Objektem je jeden (obvykle obchodní) případ, jeho několik atributů má obvykle pevnou strukturu (datum, čas, zákazník, … = identifikace košíku). Vysoký počet dalších, obvykle binárních atributů (seznam nakupovaného zboží = obsah košíku) je zadáván jako seznam atributů nabývajících nenulové hodnoty.

Asociacemi se zde rozumějí nalezené podmnožiny atributů, vyskytujících se společně (v košíku).

A	B	…	X – seznam (i >> 1]
a1	b1	…	X1 = {x1,x2,x3,x4,x5}
a2	b2	…	X2 = {x3,x8}
a3	b3	…	…
…			

Asociací zde rozumíme jeden z typů tvrzení

$$\{x1, x2, ...\} \in Xi \textbf{ se spol } V \textbf{ a podp } P \qquad (3)$$

$$\text{Je-li } A=a \wedge B=b \wedge \ldots \text{ pak } \{x1, x2, ...\} \in Xi \text{ se spol } V \text{ a podp } P \qquad (4)$$

Příklad 4.16.

transakčních asociací: Klasickým příkladem je vydolování "znalosti" z databáze o prodeji v supermarketu typu

"{jogurt, vločky} se spol 72% a podp 4567"

nebo

"je-li pátek ∧ léto, pak {pivo, buřty} se spol 67% a podp 3367"

V prvním případě se uvedená zboží umístí v prostoru co nejdále od sebe, ve druhém se v pátek vedle piva a buřtů umístí další víkendová lákadla.

♦

❑ Formulace problému

Formálně můžeme popsat náš problém takto:

Nechť $I = \{i1, i2, \ldots, im\}$ je množina prvků, D je množina transakcí T na I, kde každá transakce T obsahuje množinu prvků . Každá transakce T má jednoznačný identifikátor TID.

Hledáme asociační pravidla ve tvaru implikací

$$X \Rightarrow Y \text{, kde } X, Y \subset I \text{ a } X \cap Y = 0.$$

kde opět X nazýváme antecedentem a Y sukcedentem daného pravidla. Dále musíme opět zavést pojmy podpora a spolehlivost.

Podpora p je pravděpodobnost výskytu množiny položek v tabulce transakcí D.

Spolehlivost pravidla $X \Rightarrow Y$ je pravděpodobnost výskytu Y v transakcích, které obsahují X. Tedy

$$spol(X \Rightarrow Y) = podp(X \wedge Y) / podp(X).$$

Analýza pak spočívá v nalezení všech asociačních pravidel, které mají alespoň minimální podporu a spolehlivost, definovanou uživatelem.

Tento problém může být rozložen do dvou podproblémů:

1. Vygenerují se všechny podmnožiny X množiny I, které splňují uživatelem definovanou minimální podporu. Tyto množiny nazýváme **frekventované**.
2. Pro všechny frekventované množiny vygenerujeme pravidla, která splňují minimální spolehlivost: Pro frekventovanou množinu X a libovolné $Y \subset X$ platí,

   $$\text{jestliže } podp(X) / podp(X\text{-}Y) \geq min\_spol,$$

   pak je platné asociační pravidlo $X \Rightarrow Y$ splňující minimální podporu a spolehlivost. Tato pravidla pak nazýváme **silná.**

Podproblém 2 je jednoduše řešitelný a nemá výraznější vliv na efektivnost vyhledávacího algoritmu. Řešení podproblému 1. je zásadní pro rychlost a efektivnost a je hlavní oblastí vývoje nových a vylepšování stávajících algoritmů.

❑ Algoritmy generující transakční asociace

Pro transakční asociace je potřeba použít jiné algoritmy, než pro asociace klasické.

Nejprve uvedeme algoritmy řešící úlohu nalezení množin, které splňují podmínku minimální podpory, tedy 1. část celé úlohy.

Triviální algoritmus

Máme zformulován problém a můžeme jednoduše formulovat triviální algoritmus. Ten spočívá v tom, že generujeme všechny možné kombinace predikátů ve tvaru elementární konjunkce na levé i pravé straně pravidla, postupně prodlužujeme levou stranu a testujeme v datech, zda je výsledkem silné asociační pravidlo.

Je však zcela jasné, že takovýto algoritmus není efektivně použitelný na větších datech, kterými většinou analyzovaná transakční data jsou, jelikož bude mít exponenciální časovou složitost. Je proto nezbytně nutné nalézt jiné algoritmy, které by daný problém řešily efektivněji. Uvedeme stručné popisy některých z nich.

Apriori TID algoritmus

Jedním z řešení je varianta základního apriori algoritmu, který využívá skutečnosti, že jestliže množina **k** určitých položek není frekventovaná, pak přidáním jakékoliv položky frekventovanost položek **k+1** nemůže vzrůst (může maximálně klesat). Proto se generují kandidáti na frekventovanou množinu položek za použití frekventovaných množin nalezených v předchozím průchodu databází. Tyto kandidátské množiny jsou testovány na datech a z nich určeny frekventované množiny položek. Tak se značně omezí počet testovaných množin.

Abychom mohli formálně popsat algoritmus, je potřeba si nadefinovat některé pojmy.

- Předpokládáme, že položky v každé transakci jsou lexikograficky seřazeny a každá transakce je označena identifikátorem *TID*.
- Počet položek ve frekventované množině nazveme velikost frekventované množiny a označíme k. Množinu velikosti k pak označíme jako k-množina. Položky množiny značíme

jako c_1, c_2,..., c_k. kde $c_1<c_2<...< c_k$. Jestliže $c=X.Y$ a Y je množina velikosti m, pak Y nazýváme m-rozšířením množiny X.

- L_k je množina frekventovaných množin velikosti k. Každá položka této množiny je dvojice k-množina a její podpora.
- C_k je množina kandidátských množin na frekventovanou množinu velikosti k.
- C'_k, je množina množin kandidátských k-množin a tyto množiny kandidátských množin jsou spjaty s identifikátorem *TID* transakce ve které se nacházejí.

Algoritmus apriori TID pak můžeme popsat následovně:

První průchod algoritmem spočítá výskyt jednoprvkových množin a určí jednoprvkové frekventované množiny.

Další průchod k se pak skládá ze dvou fází. Nejprve se frekventovaná množina L_{k-1} nalezená k-1 průchodem použije pro vygenerování kandidátských frekventovaných množin C_k. Pak se prochází databáze a pro jednotlivé kandidátské množiny je spočítána podpora. Aby se urychlil tento výpočet, potřebují se rychle a efektivně identifikovat kandidáti z C_k v dané transakci. K tomu je vyvinut speciální postup.

Algoritmus aprioriTID neověřuje kandidátské množiny na databázi transakcí D, ale na množině C'_k. Prvky množiny C'_k jsou dvojice $<TID, \{X_k\}>$, kde X_k je množina potenciálních frekventovaných množin v transakci s identifikátorem *TID*. Pro k=1 je C'_1 v podstatě databáze D, kde každý prvek transakce je převeden na množinu. Pro k>1 je C'_k generována algoritmem.

Pro danou transakci t je vytvořen záznam

$<$ t.TID, $\{ c \in C_k \mid c$ je obsažena v $t\}>$.

Znamená to tedy, že pokud transakce t neobsahuje žádnou kandidátskou množinu, nebude záznam pro tuto transakci vytvořen. Díky tomu bude počet záznamů v množině C'_k menší než počet záznamů v databázi D, zejména pro vyšší hodnoty k. Navíc pro vyšší k bude velikost záznamu menší než samotná transakce, neboť transakce nebude obsahovat velké množství kandidátských množin. Avšak pro nižší k to může být naopak i větší záznam, neboť C'_k obsahuje všechny kandidátské množiny obsažené v transakci a pro malé k je větší pravděpodobnost, že transakce bude těchto množin obsahovat více.

Uvedeme si postup na příkladě.

Příklad 4.17.

Je dána množina transakcí D, je zadána minimální podpora = 2.

Spustíme tzv. ořezání se vstupem L_1 a dostaneme množinu C_2. Vypočteme podporu pro tyto kandidátské množiny pomocí průchodu množinou C'_1 a vygenerujeme množinu C'_2. První záznam v C'_1 je {{1} {3} {4}}, id transakce je 100. V kroku 7 dostaneme C_t = {{1 3}}, protože {1 3} je prvek C_2 a obě množiny ({1 3} – {1}) a ({1 3} – {3}) jsou prvky t.množiny_množin. Spustíme funkci ořezání se vstupem L_2 a dostaneme množinu C_3. Vygenerujeme množinu C'_3. Všimněme si, že neexistuje žádný záznam pro transakce 100 a 400, neboť neobsahují žádnou kandidátskou množinu z C_3. Spustíme ořezání funkci se vstupem L_4. Vrátila se nám prázdná C_4, tudíž algoritmus končí.

Databáze D

TID	položky
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

C'_1

TID	množina množin
100	{{1} {3} {4}}
200	{{2} {3} {5}}
300	{{1} {2} {3} {5}}
400	{{2} {5}}

L_1

množina	podpora
{1}	2
{2}	3
{3}	3
{5}	3

C_2

množina
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

C'_2

TID	množina množin
100	{{1 3}}
200	{{2 3} {2 5} {3 5}
300	{{1 2} {1 3} {1 5} {2 3} {2 5} {3 5}}
400	{{ 2 5}}

L_2

množina	podpora
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

C_3

množina
{2 3 5}

C'_3

TID	množina množin
200	{{2 3 5}}
300	{{2 3 5}}

L_3

množina	podpora
{{2 3 5}}	2

♦

AIS algoritmus

Kandidátské množiny jsou generovány a ověřovány „on-the-fly“ při průchodu databází. Po přečtení transakce je zjišťováno, která z frekventovaných množin nalezených v předchozím průchodu databází se nachází v dané transakci. Nové kandidátské množiny jsou generovány rozšířením těchto frekventovaných množin jinými položkami dané transakce. Frekventované množina l je rozšiřována pouze těmi položkami, které jsou frekventované a nacházejí se podle lexikografického uspořádání až za prvky množiny l. Kandidátské množiny generované z dané transakce jsou přidány do množiny kandidátských množin pro daný průběh, nebo je připočítán výskyt, pokud se již takováto množina v množině kandidátských množin nachází.

AprioriItemset algoritmus

Podstata AprioriItemset algoritmu spočívá v tom, že pro každou frekventovanou množinu je vytvořeno bitové pole, kde bit na pozici *i* reprezentuje přítomnost dané k-množiny v transakci *t.TID=i*. Generování bitového pole kandidátské k-množiny odpovídá AND operaci libovolných dvou frekventovaných (k-1)-podmnožin. Podpora pro danou kandidátskou množinu se spočítá sečtením počtu bitů nastavených na 1 na pozici *i* všech transakcí.

Algoritmus vzorkováním (Partition Algorithm)

Podstata tohoto algoritmu spočívá v rozdělení databáze D na n částí. Platí, že každá potencionální frekventovaná množina musí být frekventována alespoň v jedné části. To nám pak umožňuje nalézt frekventované množiny celé databáze pouze ve dvou průchodech všemi daty. Prvním průchodem je generována množina všech potenciálně frekventovaných množin. Tato množina je nadmnožinou všech frekventovaných množin, tzn. může obsahovat i množiny, které nakonec nebudou frekventované. Druhým průchodem je spočítána skutečná podpora pro dané potenciální frekventované množiny.

Generování pravidel

Druhá část celého problému spočívá v nalezení asociačních pravidel. Představme si proto alespoň základní princip generování těchto pravidel z frekventovaných množin.

Pro vygenerování asociačních pravidel z množiny *l* vezmeme všechny její neprázdné podmnožiny. Pro každou takovou podmnožinu *a* pak zapíšeme pravidlo ve tvaru

$$a \Rightarrow (a\text{-}1)$$

jestliže platí, že $podp(l) / podp(a) \geq minimalni\_spolehlivost$.

Tento algoritmus je možno vylepšit použitím rekurzivního generování. Například pro množinu ABCD nejprve zvažujeme podmnožinu ABC, pak AB atd. Pak, jestliže podmnožina a frekventované množiny *l* netvoří silné pravidlo, není třeba zkoušet generovat pravidla s jakoukoliv podmnožinou *a*. Je jisté, že nevynecháme žádné silné pravidlo, jelikož podpora podmnožiny *a'* množiny *a* musí být minimálně stejně velká jako podpora množiny *a*. Tudíž spolehlivost pravidla a' $\Rightarrow$ (a-1)' nemůže být větší než spolehlivost pravidla a $\Rightarrow$ (a-1).

Příklad 4.18:

Data z nákupů v supermarketu obsahují velké množství zboží. Je vhodné zboží nejprve zařadit do kategorií, případně hierarchicky uspořádaných. Například dělení v nejvyšší úrovni bude

Zboží: potraviny – drogistické – hračky – textilní – atd.
potraviny: zelenina – ovoce -mléčné výrobky – pečivo – atd.
zelenina: okurky – rajčata – brambory – atd.

Asociace je možno hledat nejen mezi jednotlivými druhy zboží na nejnižší úrovni (tam by to bylo případně včetně velikosti balení apod.), ale na vyšších hierarchických úrovních. Příkladem části výsledku může být

zboží / kategorie	**podpora**	**spolehlivost**
mléčné výrobky=>pečivo	S=0.6000	R=0.8437
jogurt => housky	S=0.2000	R=0.7500
zelenina=>pečivo	S=0.4666	R=0.9545
pečivo => zelenina	S=0.4666	R=0.6176
olej => okurky	S=0.1111	R=0.7142
olej => rajčata	S=0.1111	R=0.7142
olej => brambory	S=0.1111	R=0.7142
maso => rohlíky	S=0.2222	R=0.7692

♦

4.4. Asociace agregované

Agregovanou asociací nazveme neprůměrný vztah mezi hodnotami množiny antecedentových atributů {A, B, C…} a skupinovými ukazateli {U1, U2, …} vypočtenými z atributů sukcedentu {X, Y, …} tvaru

Je-li A=a ∧B=b ∧… , **pak** U1=u1(**V1**) ∧U2=u2 (**V2**) ∧… **s podp** *P* (5)

kde A, B, … , a, b, … mají stejný význam jako u klasických asociací
U1, U2, … jsou identifikátory definovaných agregovaných ukazatelů,
u1, u2, … vypočtené hodnoty ukazatelů pro testovanou skupinu,
V1, V2, … numericky nebo symbolicky označují, zda je Ui průměrný, statisticky významně podprůměrný nebo nadprůměrný vzhledem k základnímu souboru
P je počet výskytů skupiny neboli podpora hypotézy.

Příklad 4.19.

Porovnání výsledků asociací klasických a agregovaných

Opět použijeme data o asistované reprodukci. Skupina vygenerovaných výsledků klasickými asociacemi (použita FI, TR = transfer embryí do dělohy) je:

Jestliže		**pak**	**se spol**	**s podporou**
TR provedl gynekol =	A	výsledek gravid = netěh	82%	3251
	B	netěh	72%	2865
	C	netěh	79%	1233
	D	netěh	85%	865

Takto formulované výsledy nejsou zajímavé, ale tyto negativní výsledky (binárních hodnot) je možno interpretovat jako pozitivní:

Jestliže		**pak**	**se spol**	**s podporou**
TR provedl gynekol =	A	výsledek gravid = těh	18%	3251
	B	těh	28%	2865
	C	těh	21%	1233
	D	těh	15%	865
	…			

Průměrné procento otěhotnění po transferu je 17%. Vzhledem k různému počtu případů u různých lékařů není možné bez statistického testu rozhodnout, zda a který lékař dosahuje nad- či podprůměrných výsledků a je-li takový výsledek zajímavý.

*Vygenerovaný výsledek metodou agregovaných asociací. Použijeme ukazatel GR*100/ET, kde GR = počet gravidit, ET = počet provedených embryotransferů, průměrný GR*100/ET za celý soubor je 17%:*

Jestliže		**pak**	**s podporou**
TR provedl gynekol =	A	GR/ET = 18% -	3251
	B	28%	2865
	C	21% .	1233
	D	15% ++	865

♦

❑ Test významnosti generovaných hypotéz

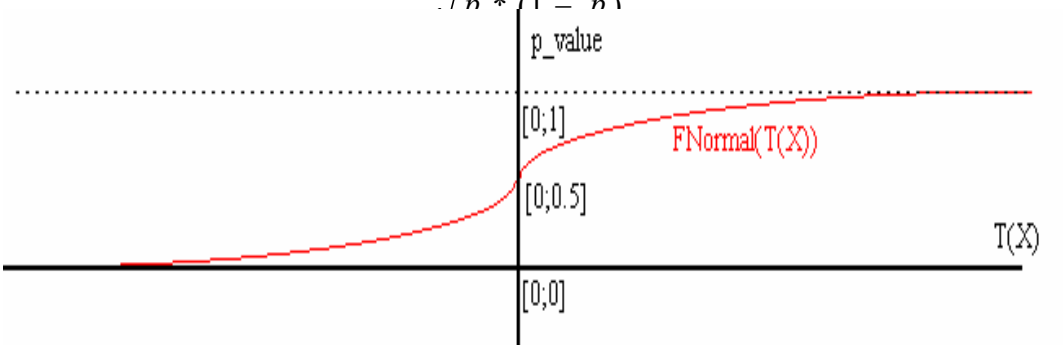
Pro otestování významnosti (neprůměrnosti) generovaných hypotéz použijeme následující výpočet pomocí hodnoty p-value.

Označíme p ... průměrná hodnota ukazatele za celý soubor

ps ... průměrná hodnota ukazatele za skupinu
ns ... počet případů ve skupině

Testovací statistika

$$T(X) = \frac{ps - p}{\sqrt{p * (1 - p)}} * \sqrt{ns}$$



Průběh normální distribuční funkce má následující graf a jeho hodnotu pro vypočítanou hodnotu T(X) dostaneme standardní procedurou FNormal.

Obr.x. Průběh hodnoty FNormal (T(X))

Výsledné rozhodnutí o statistické významnosti výsledné hypotézy zapíšeme pro uživatele snadno čitelnou jednoduchou formou. Místo hodnoty p-value, kterou by si musel inpertpretovat do slovního vyjádření podle následující tabulky:

Hodnota p-value	+podm	označení	výsledek
p-value > 0.05		=	průměr
p-value ∈ <0.05, 0.01>		.	???
p-value < 0.01 (0.001)	ps > p	+, ++	nadprůměr
p-value < 0.01 (0.001)	ps < p	-, --	podprůměr

❑ Algoritmus výpočtu asociací agregovaných

Vlastní výpočet agregovaných ukazatelů není náročnější, než výpočet kvantifikátoru u klasických asociací, proto je opět určující postup generování kombinací antecedetových atributů. Protože se opět počítají všechny ukazatele současně, je výpočet časově srovnatelný s klasickými agregacemi.

Algoritmus AGR-ASOC

1. Definují se atributy antecedentu, pro sukcedent potřebné proměnné, ukazatele a podmínky pro proměnné.
2. První průchod daty vypočte proměnné a ukazatele za celá data.

3. Generují se sentence jako kombinace hodnot antecedentu (1 kombinace = 1 skupina), k nim hodnoty proměnných a ukazatelů.

   Proměnná je definovaná agregovaná hodnota (počet, suma, ...), z níž se počítají ukazatele.

4. Každý ukazatel se otestuje na statistickou významnost pomocí hodnoty p-value.

5. Všechny výsledky nebo jen významné se uloží.

Vlastnosti algoritmu AGR-ASOC

- testuje jen kombinace hodnot skutečně přítomné v datech
- všechny ukazatele skupiny testuje jedním průchodem daty
- generuje všechny neprůměrnosti daných ukazatelů

Příklad 4.20.

Jsou dána data o asistované reprodukci o 8500 záznamech léčebných cyklů a asi 120 atributech, mj. atributech o počtech předcházejících porodů, potratů, ...o počtech odebraných a oplozených vajíček apod. Z nich expert určí charakteristické ukazatele, vypočtené ze sum a počtů případů (proměnných):

Proměnné

podmínka		
CYKL	počet léčebných cyklů AR	-
AS	počet aspirací	TC16 = 0
OO	suma odebraných oocytů OO21	TC16 = 0
OPL	suma oplozených oocytů TR26	TR26 > 0
ET	počet embryotransferů	TR29 > 0
GR	počet klinických gravidit po AR	GR > 0
POR	počet porodů po AR	GR = 9
AB	počet potratů po AR	GR = 1,2,3,4
EU	počet mimoděložních těhotenství po AR	GR = 5,6
VIC	počet vícečetných gravidit po AR	VGR = 1

Ukazatele

OO/AS	průměrný počet získaných oocytů při aspiraci
OPL/OO*100	procento oplozených oocytů ze získaných oocytů - Fertilisation rate FR
GR/ET*100	procento klinických gravidity na embryotransfer - Pregnancy rate PR
POR/GR*100	procento gravidit po AR ukončených porodem - Baby take home rate BR
AB/GR*100	procento gravidit po AR ukončených potratem
EU/GR*100	procento mimoděložních gravidit po AR
VIC/GR*100	procento vícečetných gravidit po AR

I. Jeden z úplných výsledků, kde jsou i statisticky nevýznamné výsledky pro porovnání vlivu všech hodnot antecedentu:

Věk matky

vek_k	cykl	as	oo	opl	et	gr	por	ab	eu
nevyplněno	8	8	13	4	3	0	0	0	0
<=25	1098	1037	5184	2552	484	86	43	34	9
26-30	3207	2976	13537	7046	1491	259	150	98	11
31-35	3069	2779	10850	5439	1455	237	129	91	17
36-40	1028	886	2792	1459	452	80	40	38	2
>=41	102	77	207	115	40	4	1	3	0
	8512	7763	32583	16615	3925	666	363	264	39

oo/as	opl/oo	gr/et	por/gr	ab/gr	eu/gr	vic/gr
1,62 =	30,77 =	0 =	0 =	0 =	0 =	0 =
5 =	49,23 =	17,77 =	50 =	39,53 =	10,47 =	12,79 =
4,55 =	52,05 =	17,37 =	57,92 =	37,84 =	4,25 =	15,83 =
3,9 =	50,13 =	16,29 =	54,43 =	38,4 =	7,17 =	16,03 =
3,15 =	52,26 =	17,7 =	50 =	47,5 =	2,5 =	18,75 =
2,69 =	55,56 =	10 =	25 =	75 =	0 =	0 =
	50,99	16,97	54,5	39,64	5,86	15,77

Tento výsledek je velmi neočekávaný. Říká, že věk matky nemá žádný podstatný vliv na úspěšný výsledek léčby (nikde se nevyskytují nad- nebo podprůměrné hodnoty ukazatelů). Přitom všeobecně se předpokládá opak.

II. Výsledky jen signifikantní, statisticky významné pro jeden z antecedentů:

Je-li EU1 = 0	pak	FR = 49.2 - -	s podp	6985
= 0		EU/GR = 0.8 - -		6985
= 1		FR = 56.9 + +		985
= 1		EU/GR = 25.5 + +		985
= 2		FR = 59.4 + +		472
= 2		POR/GR = 32.7 -		472
= 2		EU/GR = 14.6 +		472
= 3		FR = 59.2 +		52
= 3		EU/GR = 50.0 + +		

Závěr: *Pacientky bez EU mají nižší riziko EU a nižší FR,*

pacientky po EU mají vyšší riziko EU a vyšší FR,

vyšší počet předchozích EU zvyšuje riziko EU po AR.

♦

Shrnutí pojmů 4.

Asociace agregované, jejich tvar a rozdíl proti asociacím klasickým.

Ukazatele charakterizující agregační skupiny.

Základní asociace. Formule. Kvantifikátory. Antecedent a sukcedent. Sentence, hypotéza.

Zobecněná asociace a její tvary. Spolehlivost a podpora hypotézy.

Dedukční pravidla.

Asociace pro neúplná data.

Asociace transakční jako podmnožiny společně se vyskytujících atributů.

Nákupní košík, jeho identifikace a obsah.

Interpretace a využití výsledků transakčních asociací.

Otázky 4.

6. Jaký je rozdíl mezi klasickými asociacemi a asociacemi agregovanými?
7. Najděte v rámci látky tohoto předmětu metodu nejpodobnější výsledkům agregovaných asociací.
8. Jak by se daly výhodně zobrazit výsledky agregovaných asociací?
9. Vysvětlete podstatu jednoduchých asociací.
10. Z čeho vychází základní princip hledání asociací?
11. Jak se liší metoda hledání asociací od testování hypotéz o závislosti dvou veličin?
12. K čemu jsou prakticky užitečná dedukční pravidla pro asociace?
13. Je možno prakticky na velikých datech s neúplnou informací nalézt asociace?
14. Vysvětlete podstatu transakčních asociací a jejich rozdíl od jednoduchých asociací.
15. Pro která data se používají transakční asociace?
16. Je možno použít stejné algoritmy pro jednoduché a pro transakční asociace? Proč?

Úlohy k řešení 4.

1. Jsou opět dána data z kapitoly 5, úlohy 1 až 4. Navrhněte pro ně využití hledání agregovaných asociací: zadejte antecedenty, ukazatele, kvantifikátor a hodnoty minimální spolehlivosti a podpory. Všechny volby v zadání zdůvodněte a formulujte, co očekáváte jako možný výsledek.
2. Jsou dána data z kapitoly 5, úlohy 1 až 4. Navrhněte pro ně využití hledání asociací: zadejte antecedenty, sukcedenty, kvantifikátor a hodnoty minimální spolehlivosti a podpory. Všechny volby v zadání zdůvodněte a formulujte, co očekáváte jako možný výsledek.
3. Najděte alespoň 5 příkladů z praxe pro využití transakčních asociací, pro každý formulujte úplné zadání.

5. SHLUKOVÁ ANALÝZA

Čas ke studiu: 6 hodin

Cíl Po prostudování této kapitoly budete umět

- popsat problémy shlukovací analýzy,
- popsat typy shluků,
- popsat typy shlukovacích metod a úloh jimi řešitelných,
- pro praktické problémy rozhodnout, zda a která metoda je pro shlukování vhodná,
- pro konkrétní data provést prakticky shlukovací analýzu dat.

Výklad

5.1. Co je shlukování

❑ Klasifikace a shlukování

Shluková analýza zkoumá, zda se množina objektů **O** = {O_1, O_2 , …, O_m} zadaných reálnými atributy **A** = {A_1, A_2, …, A_n} přirozeně rozpadá na výrazné podmnožiny objektů si podobných a přitom nepodobných objektům shluků ostatních. Pokud takové podmnožiny existují, nazýváme je **shluky.**

Na první pohled jednoduchá úloha skrývá řadu problémů. Neexistuje jednoznačná definice podobnosti objektů a neexistuje ani jednoznačná definice shluku. Jak uvidíme, i zkušený analytik musí vyřešit řadu otázek souvisejících se zadanou množinou objektů, jejich atributy i důvodem shlukování, než vybere vhodnou shlukovací metodu. A stejně jako u jiných metod dolování je možno výsledky shlukování většinou formulovat jen jako hypotézy o klasifikaci zkoumaných objektů.

Shluková analýza tedy netvoří ucelenou teorii, ale skládá se z řady metod, založených na různých principech. Tato různorodost metod souvisí s různorodostí řešených problémů, požadovaných typů výsledků, časovou i prostorovou složitostí shlukovací úlohy pro velká data, neurčitostí definice shluku apod.

Než se začneme podrobněji zabývat shlukováním, zdůrazníme si rozdíl mezi různými typy rozdělování objektů do skupin:

- **klasifikací** nazveme úlohu, kdy je zadáno klasifikační kritérium (*roztřídit danou množinu lidí dle vzdělání*) nebo pravidla pro rozklad objektů,
- **shlukováním** nazýváme úlohu, kdy **kritéria klasifikace nejsou známa** nebo kdy i tato kritéria jsou předmětem našeho výzkumu; nejsou známy ani vzory objektů reprezentujících budoucí podmnožiny; pokud shluky existují, lze je interpretovat jako hledané klasifikační třídy.

Úlohy shlukování

Shluková analýza tedy řeší následující **typy úloh**:

- o množině objektů, zadaných svými atributy, není předem známo nic; úkolem je vyslovit hypotézy o jejich klasifikaci do shluků; toto je klasická úloha shlukové analýzy;
- případně dále rozpoznat, jestli existuje celá hierarchie takových rozkladů;
- pokud shluky existují, popsat, čím jsou charakteristické;
- formulovat pravidla, jak se případné další objekty zařadí do již definovaných shluků;
- jestliže je o množině objektů známo, že tvoří **k** podtříd, případně jsou známi i reprezentanti těchto podtříd, úkolem je nalézt optimální rozklad množiny do takových tříd; zde ne vždy jde o úlohu shlukování;
- pro danou množinu objektů je klasifikace předem známa; úkolem je nalézt tutéž klasifikaci algoritmicky; úloha slouží k výzkumu a ověřování samotných shlukovacích metod, případně k nalezení automatického klasifikátoru.

Historie shlukovací analýzy

Začátek oboru se uvádí do roku 1939, kdy R.C.Tryon, profesor psychologie v Kalifornii napsal první monografii "Shluková analýza".

Hlavní rozvoj oboru spadá do 60.-70. let, kdy byly publikovány mnohé monografie a články.

U nás existuje jediná monografie s roztříděním základních typů existujících obecných metod i s původními novými výsledky z roku 1985.

Průběžně se stále objevují nové algoritmy i aplikace. Aplikace užívají většinou dostupné programy, které jsou součástí velkých programových balíků.

Občas tyto programy nevyhovují, hledají se další metody, většinou si nekladou za cíl formulovat obecný algoritmus, ale jsou specializovány na užší třídu úloh (rozpoznávání objektů při analýze scény v 2D a 3D, objekty se šumem v datech, hledání podobnosti nenumerických dat apod.)

S rozvojem dalších oborů (např. neuronových sítí), se objevují i další přístupy k úloze shlukování.

S rozvojem datových skladů a dolování znalostí z nich nastává rozvoj nových algoritmů pro velmi rozsáhlá data.

Problémy úlohy shlukování

Na rozdíl od asociací nebo většiny jiných metod dolování není úloha shlukování jednoduše použitelná amatérským uživatelem. Množství existujících metod vydává různé typy výsledků. Bez jejich podrobnější znalosti může být představa uživatele o tom, co získal jako výsledek, velmi rozdílná od skutečnosti.

Než tedy přistoupíme k popisu konkrétních metod shlukovací analýzy, musíme nejprve zavést několik pojmů a formulovat, v čem jsou hlavní problémy shlukování.

Základní problémy, vznikající při řešení shlukovací úlohy jsou:

- výběr atributů charakterizujících podobnost objektů,

  měření podobnosti a nepodobnosti (vzdálenosti) objektů
  - koeficienty korelace, asociace
  - metriky
- pojem shluku a jeho geometrický model

- počet shluků rozkladu, počáteční rozklad
- pojem vzdálenosti shluků
- formulace řešené úlohy, důvod shlukování.

Postupně je probereme podrobněji.

❑ Výběr relevantních atributů a podobnost objektů

Chceme-li hledat skupiny podobných objektů v dané množině, musíme objekty nejprve dobře charakterizovat takovými **atributy, podle kterých se podobnost rozezná**. To je úloha společná pro experta a analytika.

Příklad 5.1.

Mějme data o množině lidí. Známe o nich mnoho údajů: o tělesném vzhledu – výška, váha, věk, barva pleti, očí, vlasů, ..., o jejich sportovních výsledcích – skoků, běhů, plavání, ..., o jejich studijních výsledcích – známkách z mnoha předmětů, o jejich zdravotním stavu – tlaku, rozboru krve, ... atd.

Co teď chápeme jako podobné lidi? Jsou si podobnější malí tlustí různě chytří a zdraví, nebo chytří různých pletí a sportovních výkonů, ...?

- *Známe-li o člověku jen jméno, můžeme hledat nejvýše skupiny podobných jmen, ale nebudeme nic vědět o podobnosti osob.*
- *Máme-li údaje o fyzickém vzhledu (věk, výška, váha, míry, barva očí, vlasů, typ pleti atd.), můžeme najít skupiny fyzicky si podobných lidí. Jméno nepotřebujeme, jen k identifikaci osob - stejného účinku dosáhneme přidělením jednoznačného čísla.*
- *Jiného rozdělení dosáhneme z údajů o vzdělání, prospěchu, znalostech, zájmech, výsledcích testů teoretických a praktických, výkonech. Skupiny podobných osob nebudou podobné fyzicky, ale svými schopnostmi.*
- *Prakticky nepoužitelného rozkladu dosáhneme při náhodné volbě atributů jméno, datum narození, barva vlasů a známka ze zeměpisu.*
- *Příliš mnoho aspektů – například všechny tělesné i duševní charakteristiky, jméno, adresa, adresa dědečka atd. rozdrobí osoby do velmi malých skupin a nebudou k použití.*

♦

Při výběru příliš mnoha atributů by skupin s podobnými všemi vlastnostmi bylo zřejmě příliš mnoho a příliš malých a neřešily by asi náš problém.

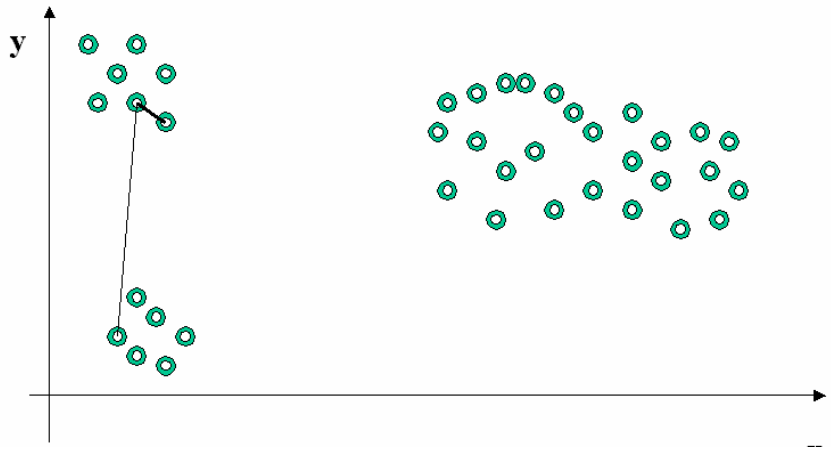
Prvním úkolem tedy je volba, **z jakého hlediska** se mají objekty klasifikovat. V souvislosti s tím se musí vybrat relevantní atributy – řekněme homogenní vzhledem ke zvolenému hledisku. Nad stejnou množinou objektů pak může vzniknout více zadání, každé nad jinou podmnožinou atributů. Výsledky spolu nemusejí souviset a pokud ano, neřeší ji shluková analýza, ale metody hledání asociací v datech.

❑ Koeficienty podobnosti a vzdálenosti

Při hledání podobných objektů vycházíme z úvahy, že dva **objekty jsou si tím podobnější, čím více atributů pro ně nabývá stejných nebo blízkých hodnot**.

Geometrický model

Pro představu o podobnosti objektů používáme geometrický model dat: každý objekt je zadán vektorem n číselných atributů. Vektor můžeme chápat jako bod v n-rozměrném prostoru a zadané objekty pak modeluje množina bodů (viz obr. x).



Obrázek 5.1. Geometrický model 2-rozměrných dat

Podobnost objektů můžeme chápat pomocí vzdálenosti bodů - čím větší vzdálenost, tím nepodobnější jsou si objekty. Dále budeme někdy místo objektů používat **body**, místo nepodobnosti pojmu vzdálenost.

Většina metod shlukové analýzy používá pojem **míry vzdálenosti** nebo naopak **míry podobnosti** objektů.

Pro vyjádření podobnosti objektů se používají buď koeficienty asociace nebo koeficienty korelace.

Pro měření vzdálenosti objektů (duální pojem k míře podobnosti) jsou používány metriky. Nejznámější a nejpoužívanější metrikou je Eukleidovská vzdálenost, ale používá se i řada dalších.

Míra vzdálenosti je používána častěji, než míra podobnosti.

❑ Měření podobnosti nebo nepodobnosti objektů

❑ Míra podobnosti objektů

Obecně pro vyjádření podobnosti objektů hledáme takový předpis, který by každé dvojici objektů (Oi, Oj) přiřadil číslo P(Oi, Oj), které bude splňovat požadavky:

$$P(Oi, Oj) \geq 0$$

$$P(Oi, Oj) = P(Oj, Oi)$$

$$P(Oi, Oi) = \max$$

Problémem může být hodnota maxima pro shodné objekty.

Předpisy používané pro vyjádření podobnosti objektů můžeme rozdělit na dva základní typy: koeficienty asociace a koeficienty korelace.

Příklad 5.2.

Pro objekty s binárními znaky může být mírou podobnosti počet shodných znaků. ♦

❑ Koeficienty asociace

jsou používány pro objekty charakterizované pouze binárními znaky. Asociaci dvou objektů Oi, Oj charakterizujeme frekvenční tabulkou tvaru

O1 \ O2	1	0	
1	a	b	r
0	c	d	s
	k	l	m

kde frekvence a, b, c, d znamenají

a počet hodnot atributů, pro které je O1(Ai) = O2(Ai) = 1
b O1(Ai) = 1 ∧ O2(Ai) = 0
c O1(Ai) = 0 ∧ O2(Ai) = 1
d O1(Ai) = O2(Ai) = 0

a, d jsou frekvence shod, b, c jsou frekvence neshod.

Koeficienty asociace jsou definovány pomocí hodnot a, b, c, d. Vyskytuje se jich v literatuře řada, většinou nabývají hodnot z intervalu <0, 1>. Uvedeme si několik z nich:

Jaccardův koeficient

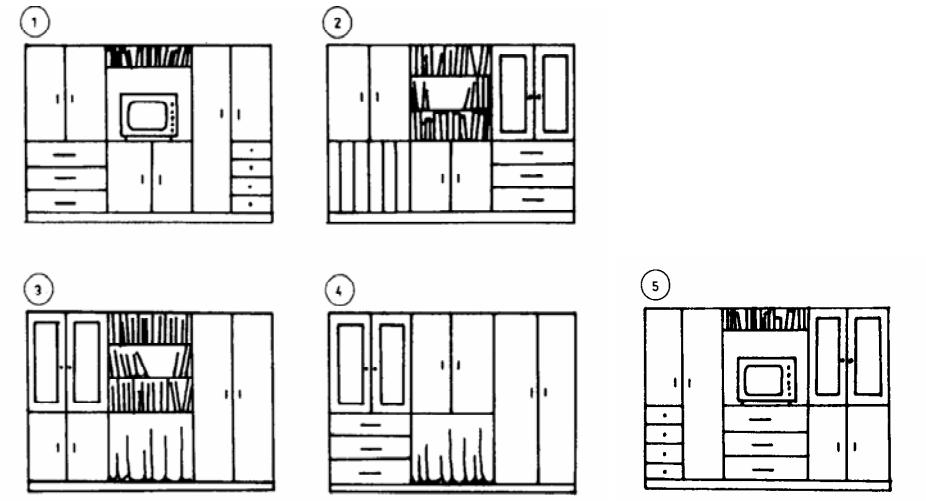
$$A_J = a / (a+b+c)$$

Sokalův koeficient

$$A_S = (a+d) / (a+b+c+d)$$

Příklad 5.3. [11]

5 skříněk bytových stěn s různým vybavením je popsáno binárními atributy, znamenajícími obsahuje/neobsahuje příslušnou skříňku. Jsou to atributy: skříňka dolní, horní, šatník, zásuvky, televizor, knihovna, prádelník, příborník, pořadač, závěs.

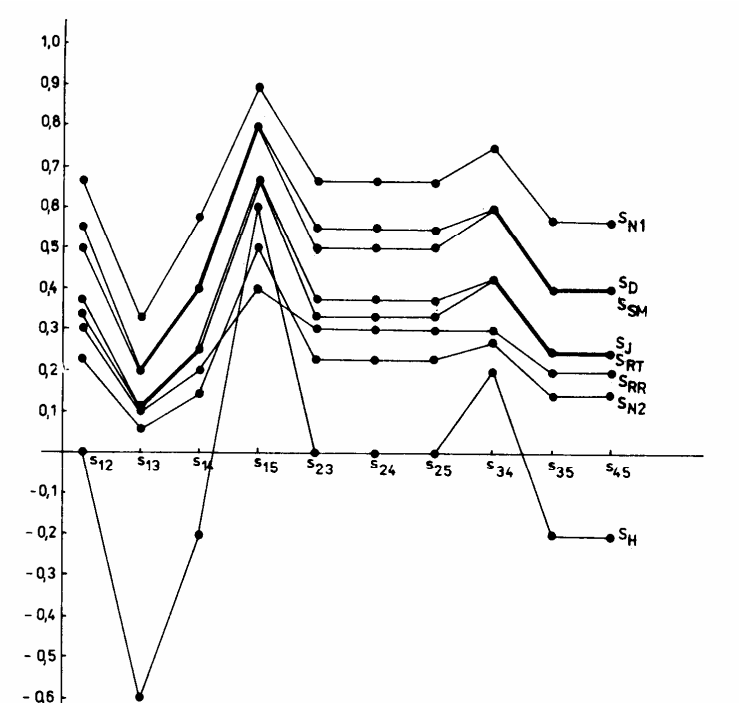


Obrázek 5.2. Skříňkové sestavy

Binární data charakterizující skříňky jsou v matici

skříň	do	hor	šat	zás	tele	kni	prá	pří	poř	záv
O1	1	1	0	1	1	0	1	0	0	0
O2	1	1	0	0	0	1	1	1	1	0
O3	1	0	1	0	0	1	0	1	0	1
O4	0	1	1	0	0	0	1	1	0	1
O5	1	0	0	1	1	0	1	1	0	0

Grafické porovnání hodnot koeficientů asociací skříněk. Vidíme, že hodnoty koeficientů se sice liší svými absolutními hodnotami, ale jejich vzájemné vztahy se téměř neliší.



Obrázek 5.3..Grafické porovnání asociací objektů různými koeficienty asociace

♦

❑ Koeficient korelace

Koeficient korelace objektů Oi = (x_{i1}, . . . , x_{in}), Oj = (x_{j1}, . . . , x_{jn}) s reálnými atributy se určí podle vzorce

$$r_{ij} = \frac{k_{ij}}{s_i \,.\, s_j} \qquad k_{ij} = \frac{1}{n}\sum_{l=1}^{n} x_{il} \,.\, x_{jl} - \bar{x}_i \,.\, \bar{x}_j$$

kde $\underline{x}_i$ a $\underline{x}_j$ jsou střední hodnoty objektů Oi, Oj,

s_i a s_j jsou směrodatné odchylky objektů.

Tento koeficient je použitelný jen v některých případech pro reálné znaky, kdy mají všechny stejnou měrnou jednotku, nebo po standardizaci. Jinak průměr hodnot znaků jednoho objektu nemá reálný smysl.

❑ Míra vzdálenosti objektů

Duálním pojmem k míře podobnosti objektů je míra nepodobnosti neboli **míra vzdálenosti** objektů.

Pro měření vzdálenosti objektů jsou používány metriky.

Metriky vychází z geometrického modelu dat, kde objekty o n znacích chápeme jako body v n-rozměrném Euklidovském prostoru $\mathbf{E_n}$. Pak podobnost objektů můžeme vyjadřovat pomocí vzdálenosti jim odpovídajících bodů.

Metrika V je funkce definovaná na $\mathbf{E_n \times E_n}$, která přiřazuje každé dvojici bodů (Oi, Oj) číslo V (Oi, Oj) takové, že platí:

$$V(Oi, Oj) = 0 \Leftrightarrow Oi = Oj$$

$$V(Oi, Oj) \geq 0$$

$$V(Oi, Oj) = V(Oj, Oi)$$

$$V(Oi, Oj) + V(Oj, Ok) \geq V(Oi, Ok)$$

Některé z používaných metrik:

Eukleidovská vzdálenost - nejznámější metrika

$$\mathbf{V(Oi, Oj) = \sqrt{\left(\sum_{i=1}^{n} (ai - bi)^2\right)}}$$

Metrika V1 daná předpisem

$$\mathbf{V1(Oi, Oj) = \sum_{i=1}^{n} |ai - bi|}$$

Sokalova metrika

$$\mathbf{V_S(Oi, Oj) = Ö\,(V^2(Oi, Oj) / n)}$$

Sup - metrika V_∞

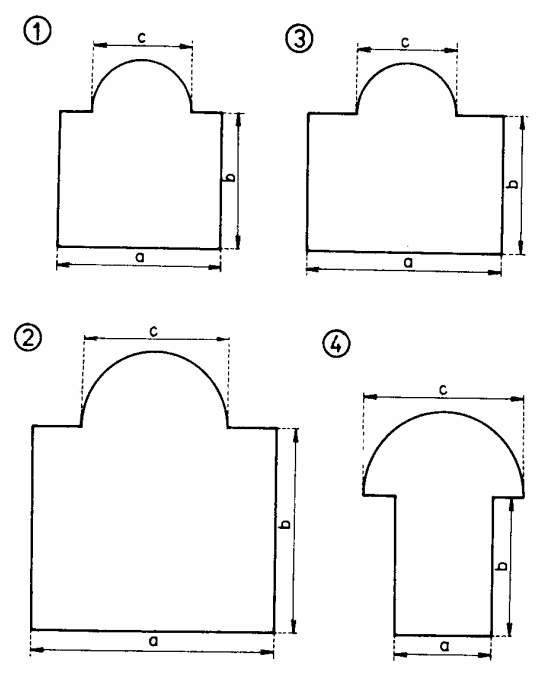
$$\mathbf{V_\infty(Oi, Oj) = \max_{i=1,\ldots,n} \{|ai - bi|\}}$$

Příklad 5.4. [11]

*Máme 4 jednoduché obrazce složené z obdélníka o stranách **a, b** a půlkruhu o poloměru **c**.*

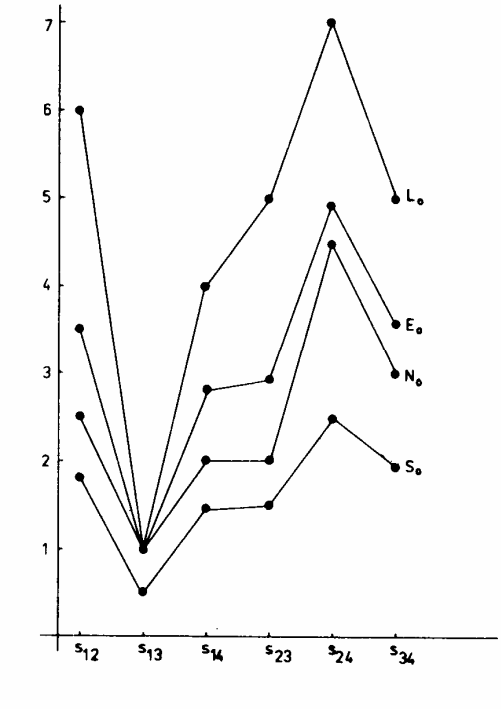
Jejich popis v datové matici je

obraz	a	b	c
O1	5.0	4.0	3.0
O2	7.6	6.0	4.4
O3	6.0	4.0	3.0
O4	3.0	4.0	5.0



Obrázek 5.4. Shlukované obrazce

Grafické porovnání hodnot různých metrik. Opět vidíme, že hodnoty se sice liší svými absolutními hodnotami, ale jejich vzájemné vztahy se téměř neliší.



Obrázek 5.5. Grafické porovnání vzdáleností objektů různými metrikami

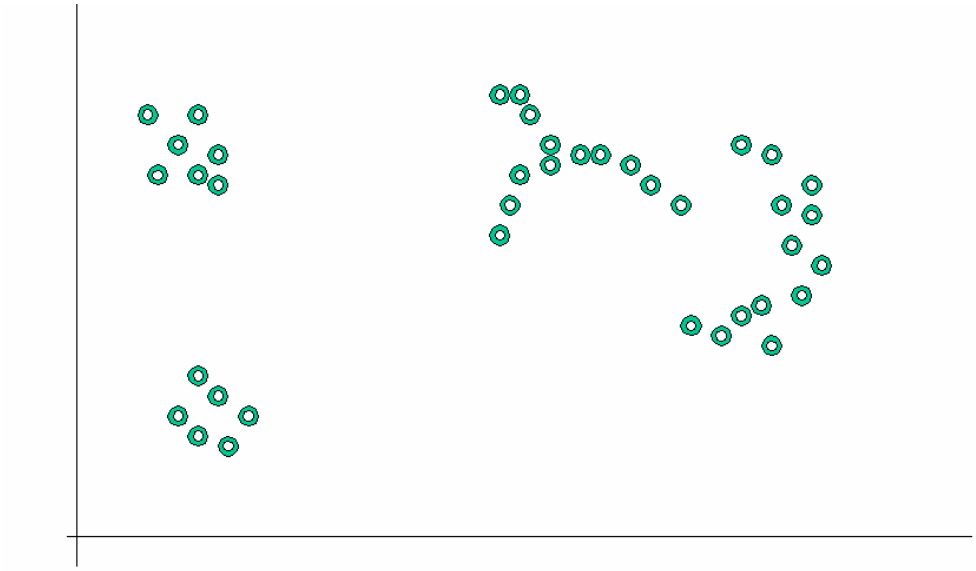
♦

Pojem shluku

Definovali jsme podobnost a vzdálenost dvou objektů, dále je nutné definovat, co je shluk objektů. V literatuře se často shluk uvádí jen vágním popisem, zachycujícím intuitivní představy analytika.

Příklad 5.5.

Podívejme se na několik bodů v rovině na obr. x, které chápeme jako obrazy dvourozměrných objektů s reálnými atributy. O dvou skupinách v levé části obrázku se zřejmě všichni shodnou, že tvoří shluky. Intuitivně chápeme jako shluk body blízko sebe a poměrně vzdálené bodům ostatním. Ovšem body v pravé části obrázku již tak jednoznačné nejsou. Tvoří dva „rozsochaté" útvary, ale není možné o nich říct, že v rámci jednoho útvaru jsou všechny blízko a vzdálené bodům útvarů ostatních. Jsou to tedy shluky nebo ne?



Obrázek 5.6. Skupiny bodů v rovině

Z mnoha definicí shluků v literatuře uváděných se uvedeme jako příklad dvě, reprezentující tato rozdílná chápání pojmu shluk.

Definice 5.1.

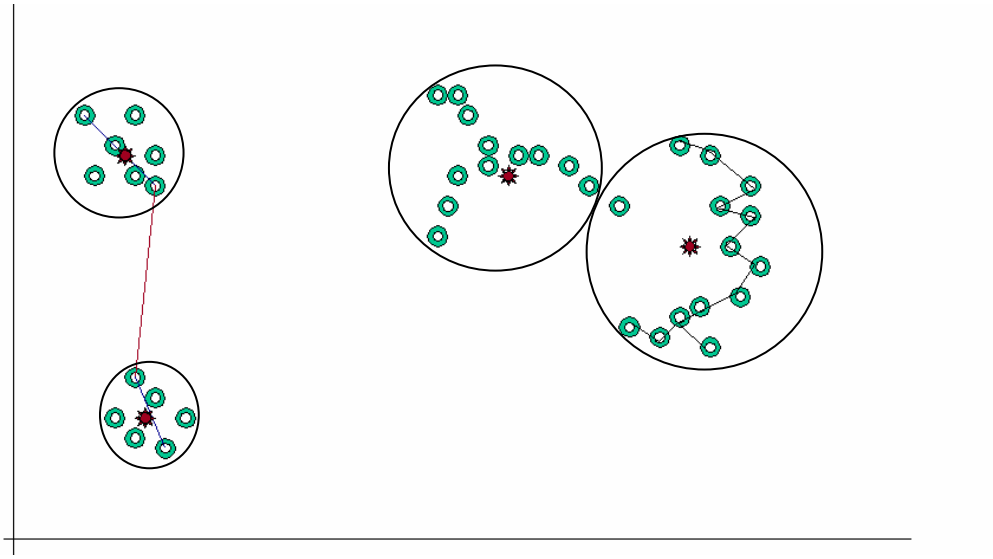
Je dána množina objektů O = {O1, . . . , Om} a koeficient vzdálenosti objektů V. Shlukem nazveme takovou podmnožinu X ∈ O, pro niž platí

$$\max_{Oi,Oj \in X} V(Oi, Oj) < \min_{Oi \in X,\ Ok \notin X} V(Ok, Oi)$$

Definice 5.2.

Je dána množina objektů O = {O1, . . . , Om} a koeficient vzdálenosti objektů V. Objekt Ox nazveme α-souvislým s objektem Oy pro daný práh α, existuje-li řetěz objektů Ox = O1,O2,...,Ok = Oy, k > 1, takový, že V (Oi, Oi+1) ≤ α pro i = 1, . . . , m -1.

α-souvislým shlukem (**α-shlukem**) nazveme takovou podmnožinu X ∈ O, pro niž platí, že každý pár objektů z X je α-souvislý a žádný objekt z (O – X) není α-souvislý s žádným objektem z X.



Obrázek 5.7. Skupiny bodů tvořící shluky 2 různých typů

Definici 1 odpovídají skupiny bodů vlevo, ale ne dvě skupiny bodů vpravo. Všechny čtyři skupiny ale odpovídají definici 2.

Typy shluků

První typ shluku nazveme **kulovým** – protože jeho body jsou rozloženy okolo přirozeného středu shluku, okolo jeho těžiště.

Druhý typ shluku může nabývat libovolný „tvar“, za shluk jej považujeme proto, že existuje mezi každými dvěma objekty „cesta“, spojující sousední objekty vzdáleností dostatečně malou. Tyto shluky nazýváme **přirozenými** nebo obecnými.

Zřejmě kulové shluky jsou zvláštním případem shluků přirozených. Obecně nelze vyžadovat, aby shluky byly kulové, buď jsou, nebo nejsou. Reálné shluky mohou existovat v nejrůznějších seskupeních.

Jak uvidíme níže, ne všechny shlukovací metody chápou shluk ve stejném smyslu, a proto i jejich výsledky nad stejnou množinou objektů jsou různé. Našim úkolem bude umět se orientovat jednak v tom, jaký výsledek potřebujeme, jednak v tom, kterou metodu k jeho dosažení máme zvolit.

Existují metody, které najdou přirozené shluky jakýchkoliv tvarů. Jsou však také metody, které dávají jako výsledky „shluky“ vždy kulové, ať je skutečnost jakákoliv (viz. obr. 5.7., kde se bod patřící k jednomu přirozenému shluku dostal do jiného kulového „shluku“).

Ovšem i toto řešení může být někdy užitečné – pokud hledáme optimální rozložení bodů vzhledem k několika středům. Není to však úloha shlukovací. Ještě se k tomuto problému vrátíme u konkrétních metod.

Charakteristiky shluků

Konečným cílem při shlukování obvykle je nalézt pravidla, podle kterých se objekty zařazují do jednotlivých nalezených shluků. K tomu potřebujeme každý shluk popsat nějakými údaji, které by shluk vhodně charakterizovaly. Které údaje to mohou být?

Přirozeně by se nabízelo těžiště shluku, tedy bod, jehož atributy tvoří průměrné hodnoty atributů bodů shluku. Ovšem již z jednoduchého obr. 5.7. je vidět, že **těžiště není vhodná charakteristika shluku.** Někdy těžiště dokonce neleží „uvnitř“ shluku.

Jako další údaje se nabízí vzdálenosti vnitroshlukové (co nejmenší) a mezishlukové (co největší), jak by odpovídalo definici 1. Ovšem z obrázku je patrno, že to také nejsou vhodné charakteristiky.

Optimálně by měla definice shluku (a tedy i charakteristika kvality rozkladu a charakteristika příslušnosti objektu ke shluku) brát v úvahu současně

- vzájemnou podobnost bodů uvnitř shluku (minimální)
- vzájemnou vzdálenost shluků navzájem (maximální)
- tvar shluků (kulový – přirozený tvar)
- charakteristické vlastnosti jednotlivých shluků:

  počet bodů, těžiště, minimum, maximum, stand. odchylky, ..,, homogenita
- charakteristické vlastnosti celkového rozkladu na shluky:

  průměrná minimální vzdálenost shluků

  průměrná minimální vzdálenost těžišť

  vzájemné vzdálenosti těžišť

  celková Σ čtverců chyb

Taková charakteristika ovšem neexistuje, protože by musela obsahovat i protichůdné parametry. Proto neexistuje ani jednoznačný výsledek. Později si uvedeme, jakými údaji je možno shluky popsat pro uspokojivý popis.

❑ Typy shlukovacích metod

Již jsme zmínili, že shlukovou analýzu tvoří řada metod, založených na různých teoriích nebo jen na intuitivních představách autorů, jak se k vágnímu pojmu shluk dobrat. Do každého algoritmu je zabudována jakási „heuristická“ zkušenost konstruktéra metody, různé metody to provádí různou taktikou. Přesto mnozí autoři nezávisle na sobě definovali velmi podobné algoritmy a tak chápali „shluk“ stejně.

Existující metody je možno dělit podle několika kritérií, nejdůležitější jsou

dle cíle shlukování na

- **nehierarchické**, produkující prostý rozklad objektů na podmnožiny;
- **hierarchické**, produkující hierarchii rozkladů, kde každý rozklad je zjemněním předcházejícího; jsou vyvinuty různými autory, liší se většinou definicí podobnosti objektů a podobnosti shluků.

dle tvaru výsledných shluků v geometrické interpretaci na

- shluky **kulové**, body soustředěné víceméně pravidelně kolem svého těžiště,
- shluky **obecné** tvoří souvislé husté oblasti nejrůznějších tvarů, které není možno dost dobře charakterizovat těžištěm.

dle rozložení výsledných shluků na

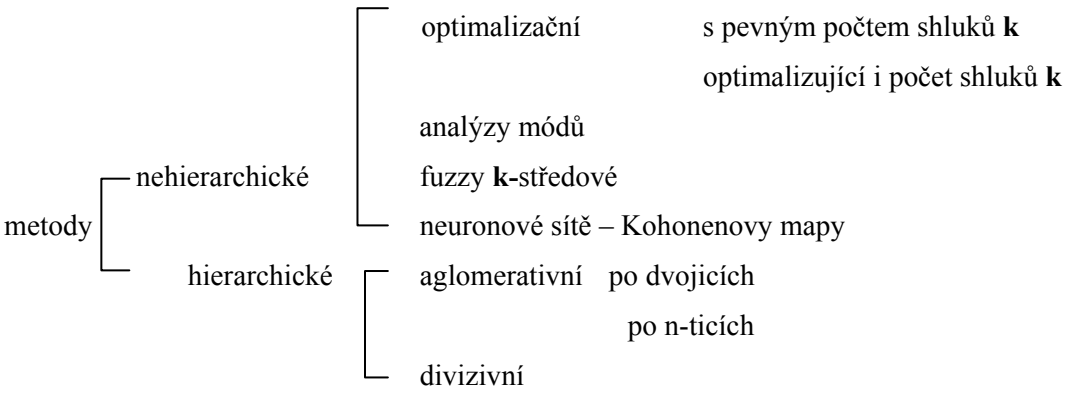
- **disjunktní** shluky (klasické shlukování, odpovídající nepodobnosti objektů různých shluků)
- překrývající se shluky nebo též **fuzzy-shluky** pro zvláštní případ zadané úlohy.

dle procesu shlukování (typu algoritmu)

- **sekvenční** - množinu objektů jedním (sekvenčním) průchodem rozdělují do shluků; obvykle se používaly pro velký počet objektů, které se nevešly do paměti, výsledky nebývaly kvalitní;
- **paralelní** - množina objektů je podle potřeby procházena vícekrát
- **přímé** - algoritmus přímo vytváří hledaný rozklad, může procházet objekty vícekrát;
- **optimalizační** - hledají optimum nějaké kriteriální funkce, obvykle iteracemi z nějakého zadaného či vygenerovaného počátečního rozkladu; průběžně je vytvářena řada zlepšujících se rozkladů;
- **rekurzivní** - hierarchické, které vytvářejí následující rozklad pomocí předcházejícího rozkladu, ne pomocí původních objektů;
- **heuristické** - v jistém smyslu všechny, zde však chápeme jako heuristické ty, které využívají k hledání rozkladu prohledávání prostoru všech možných rozkladů s využitím heuristické funkce.
- **neuronovou sítí** - proces nealgoritmický, při vhodném počátečním nastavení sítě rychlý,

Tomuto dělení odpovídají různé typy definic pojmu shluk. Každá metoda umí vyhledat jeden typ výsledku – tvaru a rozložení shluků, obvykle se to však v literatuře nijak nezdůrazňuje. Pro volbu metody vzhledem k dané úloze nebývají uvedena kritéria, vše záleží na intuitivní volbě či zkušenostech analytika. Také u nabízených SW systémů pro dolování dat nebývají tyto skutečnosti (zvláště tvar výsledných shluků) nijak uvedeny a nedostatečně poučený uživatel může dostat velmi zkreslené výsledky, aniž o tom ví.

Nejrozšířenější typy metod je možno rozdělit dle typu (hierarchie/ne), typu rozkladu (disjunktní/ne), částečně dle tvaru výsledku (kulový/přirozený) na následující metody. Mimo to mohou existovat mnohé další metody, které obvykle řeší specifické úlohy.

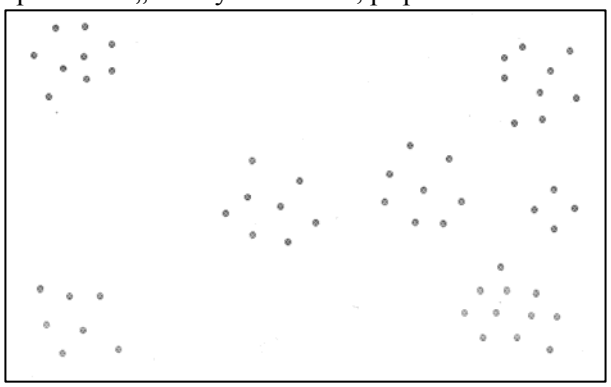


Charakteristiku uvedených typů a jejich základní algoritmy uvedeme postupně podrobně.

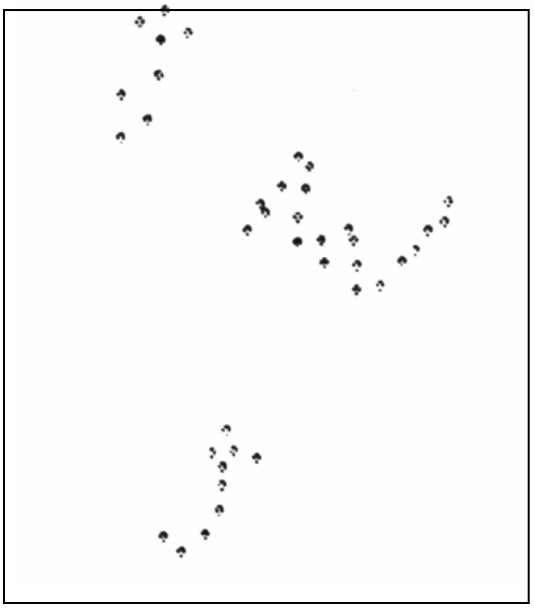
❑ Testovací data pro shlukování

V následujících kapitolách budeme používat pro prezentaci výsledků jednotlivých metod několik typů dvourozměrných dat. Data byla vytvořena pro testovací účely. Dvourozměrná jsou proto, že jejich zobrazením v rovině s vykreslením výsledku se snadno porovná „intuitivní“ představa uživatele o výsledku se skutečným výsledkem každé metody.

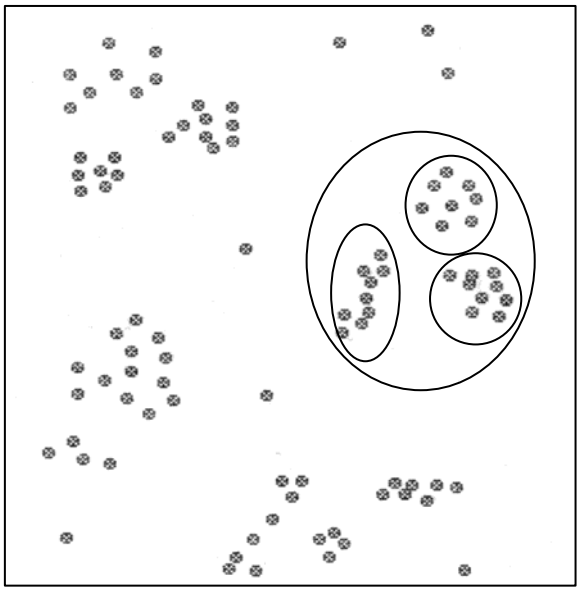
Data 1: Sedm zřetelných přirozeně „kulových“ shluků, případně 2 úrovně shlukování



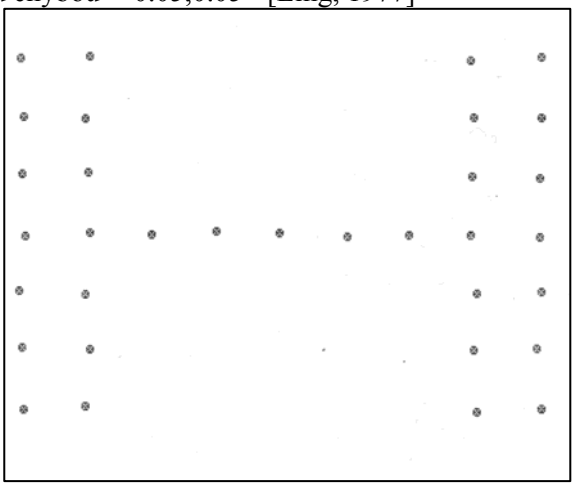
Data 2: Tři zřetelné obecné sh' '



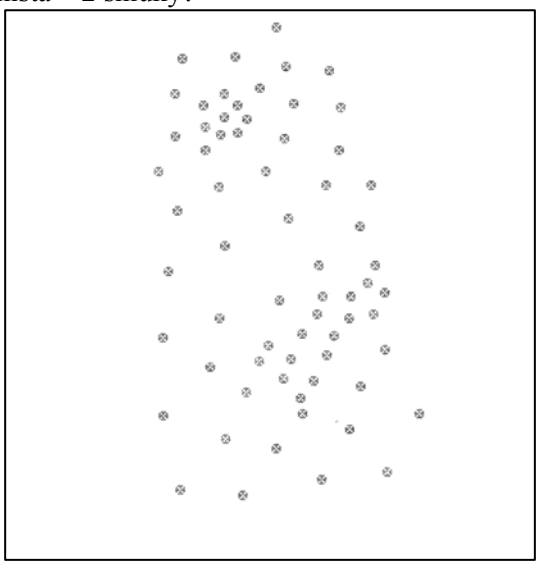
Data 3: shluky ve 2 shlukovacích úrovních + izolované body



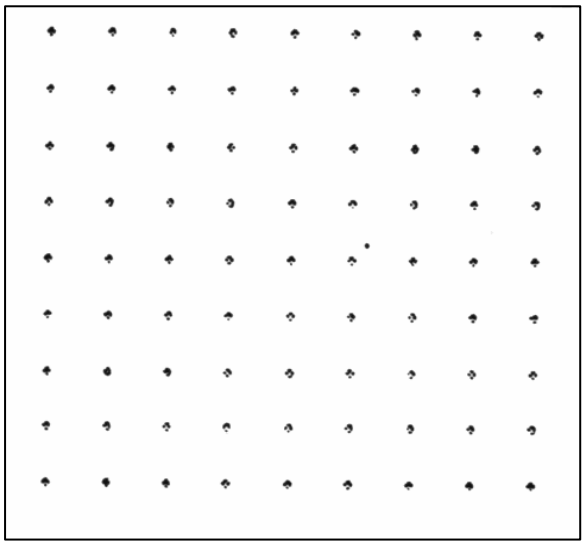
Data 4: H-data s náhodnou chybou <-0.05,0.05> [Ling, 1977]



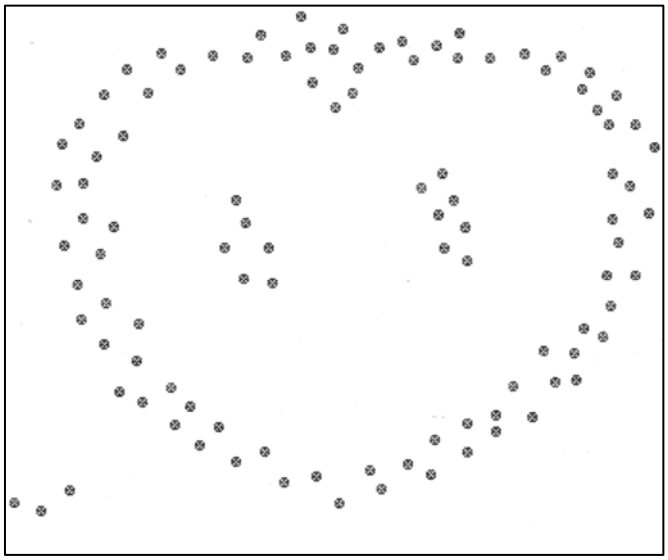
Data 5: 2 nevýrazná hustší místa ~ 2 shluky?



Data 6: homogenní data bez shluků = 1 shluk



Data 7: „Srdce“ se třemi malými shluky uvnitř i vně



Shrnutí pojmů 5.1.

Klasifikace a shlukování. Úlohy shlukování.

Problémy úloh shlukování.

Výběr relevantních atributů a podobnost objektů.

Koeficienty podobnosti a vzdálenosti.

Míra podobnosti objektů. Koeficienty asociace. Koeficient korelace.

Míra vzdálenosti objektů.

Pojem shluku. Typy shluků.

Shluky kulové a shluky přirozené.

Charakteristiky shluků.

Typy shlukovacích metod.

Metody hierarchické a nehierarchické. Shluky disjunktní a shluky překrývající se.

Otázky 5.1.

4. Co rozumíme klasifikací objektů a co shlukováním?
5. Které úlohy musíme postupně vyřešit při shlukování objektů?
6. Podle čeho vybíráme atributy vhodné pro popis shlukovaných objektů?
7. Jak měříme podobnost objektů a které míry podobnosti znáte?

8. Co je míra vzdálenosti objektů, jak souvisí s podobností a k čemu se používá?
9. Které míry vzdálenosti objektů znáte?
10. Co je shluk?
11. Jaké typy shluků rozlišujeme?
12. Čím je možno shluky charakterizovat?
13. Které hlediska pro dělení a které typy shlukovacích metod znáte?
14. Jaké typy výsledků dostáváme metodami nehierarchickými a metodami hierarchickými?

Úlohy k řešení 5.1.

1. Jsou dána data BANKA, obsahující 1027 záznamů o poskytovaných úvěrech za minulých 5 let. Jejich struktura je

   BANKA pohlavi [muž / žena],
   vek [roku],
   svobodny [ano / ne],
   nezamestnany [ano / ne],
   cim_ruci [auto, dum, PC, kolo],
   problematicky_region [ano / ne],
   mes_prijem [Kc],
   hotovost_u_banky [Kc],
   pocet_mesicu_splatky,
   pocet_roku_u_souc_firmy,
   dostal_uver [ano / ne],
   splacel_bezproblemově [ano / ne])

   Formulujte shlukovací úlohu (jak se dá využít shlukování nad těmito daty), vyberte z dat vhodné atributy, navrhněte metody jejich předzpracování, míru podobnosti či vzdálenosti a určete typ hledaných shluků.

2. Jsou dána data z evidence studentů sportovního GYMNÁZIA s atributy: školní rok, třída, učitel [jméno], jméno [studenta], pohlaví [chl/dív], věk, výška, váha, dále maximální výkony sportovní za aktuální rok ve skoku vysokém [cm], dalekém [cm], běhu_100m [12.3 sec], běhu-400m a závěrečné známky z češtiny, cizího jazyka [ruština a později angličtina], matematiky, fyziky, dějepisu a zeměpisu. Data jsou pořizována za dobu 30 let.

   Formulujte shlukovací úlohu (jak se dá využít shlukování nad těmito daty), vyberte z dat vhodné atributy, navrhněte metody jejich předzpracování, míru podobnosti či vzdálenosti a zvolte typ hledaných shluků.

5.2. Shlukování nehierarchické

Cíl Po prostudování této kapitoly budete umět

- rozlišit úlohu optimálního rozkladu od úlohy shlukovací,
- vybrat vhodnou metodu pro výpočet dané úlohy a nastavit její vstupní parametry,
- pro praktické problémy rozhodnout, zda a která metoda je pro shlukování vhodná.

Výklad

Metody nehierarchické hledají nejlepší prostý rozklad množiny objektů na shluky, tedy na disjunktní podmnožiny. Řada typů těchto metod dává různé typy výsledků.

Počet možných rozkladů m objektů na podmnožiny je dán rekurzivně vztahem

$$P[1] = 1 \qquad P[n+1] = 1 + \sum_{i=1}^{n} \left(\binom{n}{i} . P[i] \right)$$

výraz konverguje k

$$1 + \Sigma \left(P[n] * x^n \right) / n! \quad \rightarrow \quad \exp(\exp(n) - 1)$$

Příklad 5.6.

pro n =	10	*je počet podmnožin*	1.16 E 5
	20		5.17 E 13
30		8.74 E 23	
40		1.57 E 35	
	...		

♦

Odtud je zřejmé, že není možno procházet a testovat všechny možné rozklady, je nutné hledat algoritmy testující jen „pravděpodobnější“ rozklady, tedy **do algoritmu je zabudována zkušenost konstruktéra metody**. Různé metody to provádí různou taktikou.

❑ Metody optimalizační

hledají nejlepší rozklad množiny objektů iteračním způsobem. Počáteční rozklad (zadaný nebo vygenerovaný) zlepšují tak, že hledají rozklad s lepší hodnotou zadané kriteriální funkce. Metody se navzájem liší způsobem prohledávání množiny všech možných rozkladů. Jednodušší z nich hledají nejlepší rozklad pro daný počet shluků, obecnější hledají současně nejlepší rozklad i počet shluků.

Cílem je najít takový rozklad, pro který kriteriální funkce nabývá extrému. Po zvolení kriteriální funkce je úloha teoreticky definovaná a je možno optimální rozklad najít. Problémem je zvolit metodu, která vede rychle k cíli. Prohledávat všechny možné rozklady, byť jich je konečně mnoho, často není únosné ani na počítačích - pro jejich velký počet. Ovšem ani formulování kriteriální funkce není tak snadné, proto vzniklo mnoho metod.

Optimalizační metody vycházejí **z předem určeného počtu shluků** a iteračně zlepšují rozklad vzhledem k zadanému tvaru kriteriální funkce nebo k jinak definovanému konci algoritmu (někdy se tyto typy algoritmů nazývají horolezeckými).

❑ Problém počátečního rozkladu

Algoritmy optimalizační začínají zadáním nebo odvozením počátečního rozkladu množiny objektů. Optimální je případ, kdy uživatel má nějakou znalost nebo představu o datech předem, dovede určit počet shluků i jejich přibližné charakteristiky. Pokud ne, používá se buď náhodné zadání počátečních bodů, nebo se provádí jednoduché předzpracování, které rozmístí počáteční body rovnoměrně do zadané množiny bodů.

Počet bodů = počet shluků označíme **k**. Také tento počet uživatel buď zadává, nebo se provádí několik výpočtů s různými **k** a vybere se ten s nejlepší hodnotou kriteriální funkce.

Metody zadání počátečního rozkladu

TYP1 prvních **k** bodů zadané množiny dat

TYPN **k** bodů náhodně vybraných ze zadaných

body náhodné se souřadnicemi náhodně vygenerovanými v rámci variačních intervalů souřadnic

TYPD body vybrané uživatelem z množiny daných bodů

body zadané uživatelem pomocí souřadnic (hodnot atributů)

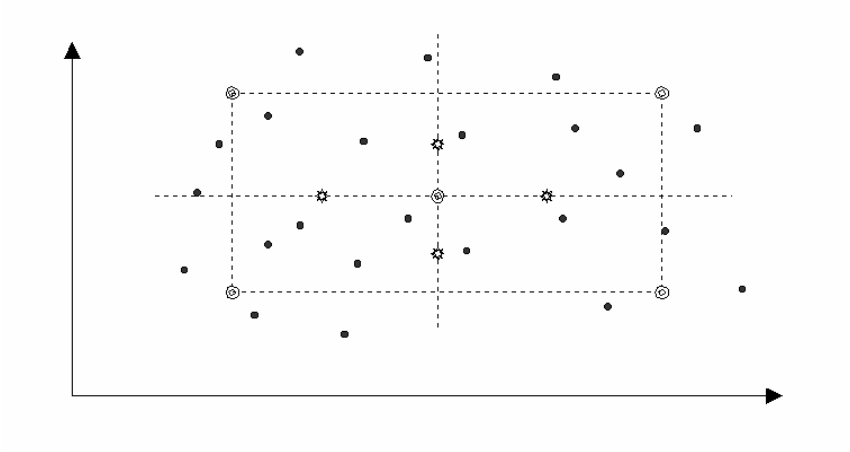
TYPF (Fromm)

$k=1 + 2^s$ bodů v těžišti souboru a ve vrcholech s-rozměrného kvádru se středem v těžišti a stěnami rovnoběžnými se souřadnými rovinami a hranami rovnými dvojnásobku standardní odchylky příslušné souřadnice ...

$$y_{j\,i} = t_i + \mathrm{sgn}\left(\sin\left(2.\pi / 2^i - 2.\pi / 2^{i+1}\right)\right) s_i$$

TYPC

k bodů v těžišti a v bodech posunutých od těžiště v kladném a záporném směru podél jedné z prvních souřadných os o standardní odchylku příslušné souřadnice.



Obrázek 5.8. Analýza nasazení počátečních typických bodů dat

❑ Metody optimalizační k-středové s pevným počtem shluků

Nejužívanější „shlukovací“ metodou, vyskytující se ve většině SW systémů pro statistiku, analýzy dat i pro dolování znalostí, je k-středová metoda se zadaným pevným počtem shluků. Existuje několik jejích variant, které se jen málo liší iteračním algoritmem. Protože jde o velmi jednoduchý, přirozený a relativně rychle konvergující algoritmus, je jeho obliba vysoká.

První základní verzi formuloval Forgy.

Základní algoritmus FORG

1. zadání počtu shluků **k**,
2. zadání **k** počátečních typických bodů,
3. přiřazení každého bodu k nejbližšímu typickému bodu a jemu odpovídajícímu shluku,
4. výpočet těžiště každého z **k** shluků,
5. definování nových typických bodů ve vypočtených těžištích,
6. pokud došlo ke změně v přiřazení bodů shlukům, opakování od bodu 2.,
7. výpočet charakteristik výsledného rozkladu.

Od tohoto algoritmu jsou pak odvozeny některé varianty, které však nemají výrazně odlišný ani průběh výpočtu, ani výsledek.

Varianta Janceyova

JANC

algoritmus jako u FORG, jen bod 4 definuje nové typické body jinak:

4. definování nových typických bodů do bodů souměrně sdružených s původními typickými body podle těžiště

Varianta MacQueenova

MACQ

od předcházejících dvou metod se liší tím, že přepočítává typický bod skupiny po každém přemístění bodu. To způsobuje závislost výsledného rozkladu na uspořádání množiny shlukovaných objektů. Tato metoda provádí přiřazení jen 2x, neiteruje rozklady až do ustálení bodů.

1. zadání **k** počátečních typických bodů,
2. přiřazení každého bodu k nejbližšímu typickému bodu a přepočtení typických bodů zmenšené i zvětšené skupiny.

Varianta Wishartova

WISH

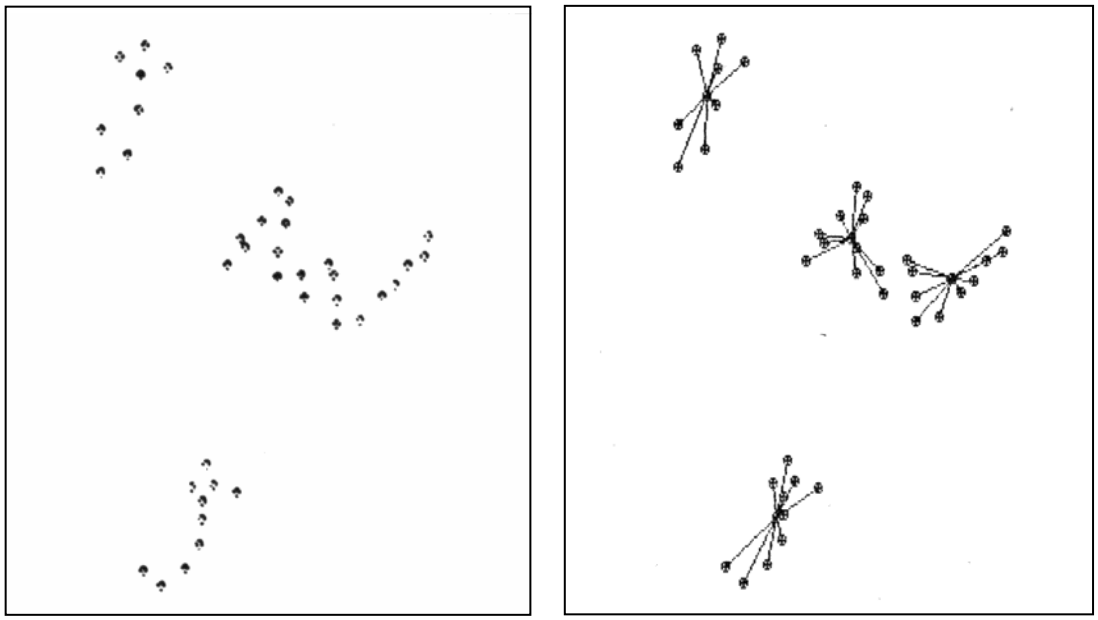
konvergentní varianta metody MacQ, po dvou bodech algoritmu MACQ následují

3. pokud došlo ke změně v přiřazení bodů shlukům, opakování od bodu
4. výpočet kriteriální funkce výsledného rozkladu

Protože jde o velmi rozšířené metody, podíváme se na jejich výsledky podrobně na testovacích datech.

Příklad 5.7.

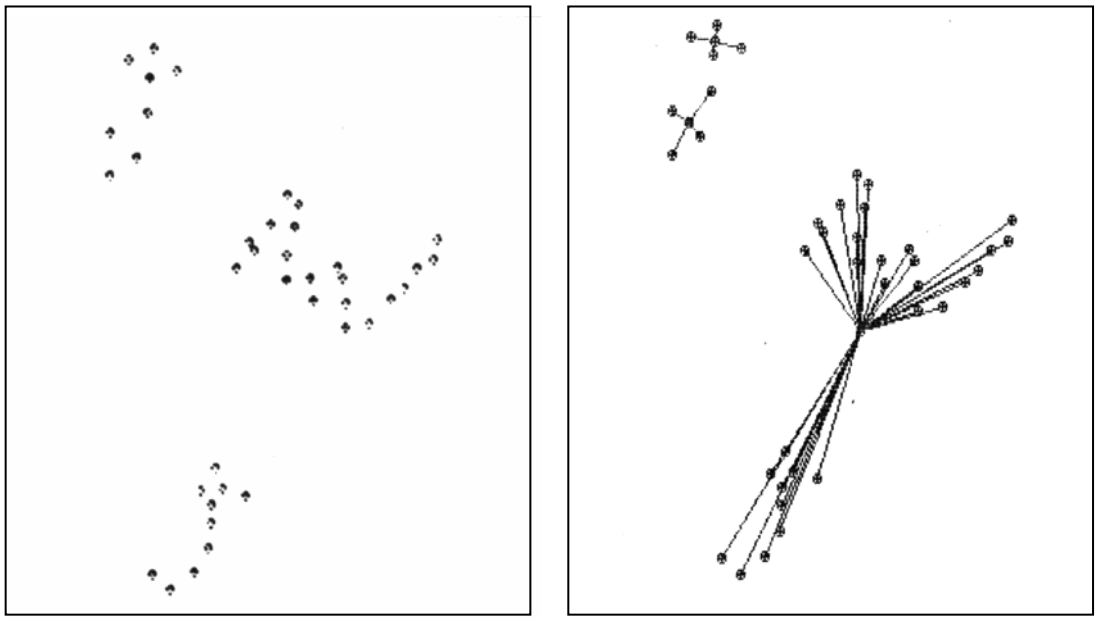
3 shluky, metoda k-středová, 4 typické body



♦

Příklad 5.8.

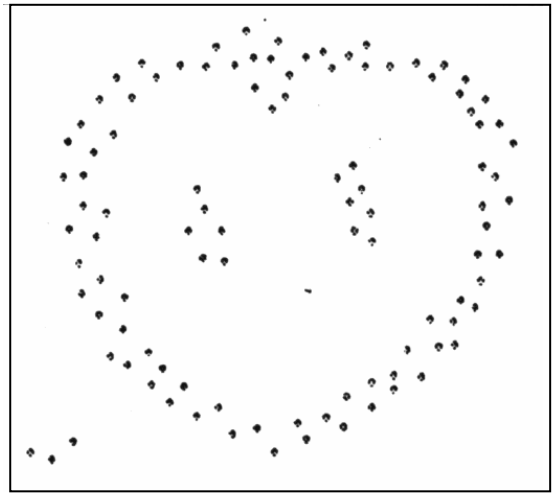
3 shluky, metoda K-středová, 3 náhodné typické body

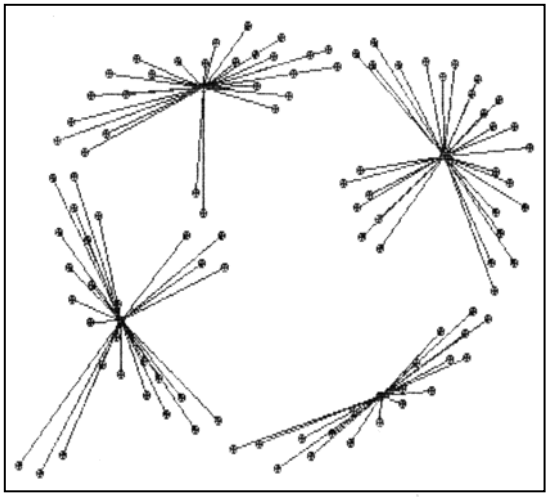


♦

7Příklad 5.9.

4 obecné shluky, metoda K-středová, 4 odvozené typické body

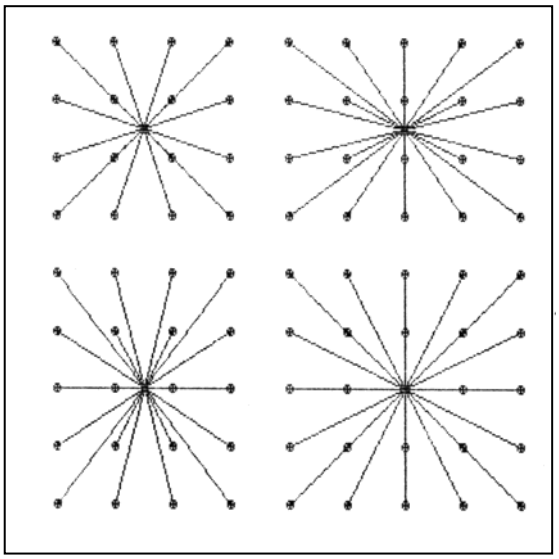


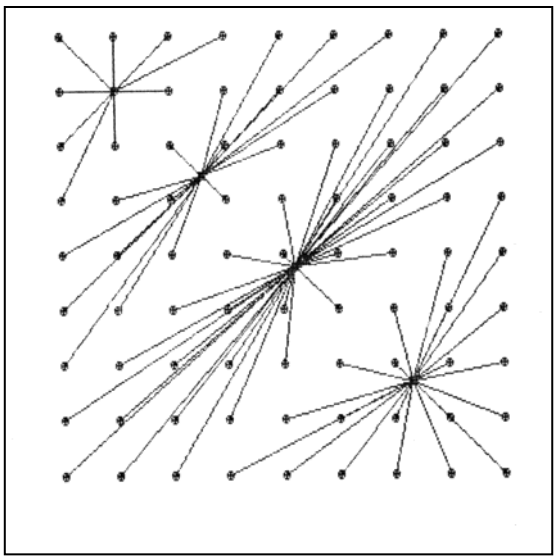


♦

Příklad 5.10.

homogenní množina, metoda K-středová, 4 náhodné typické body (2 x jiné)





♦

Závěr: metoda k-středová neshlukuje, výsledkem nejsou přirozené shluky, ale **k** kulových podmnožin objektů. Metoda nerozpozná, zda množina bodů skutečně obsahuje shluky a kolik, ale vykáže jako výsledek „shluky“ i v dokonale homogenní množině. Je vysoce závislá na počáteční volbě typických bodů vzhledem ke shlukované množině bodů a někdy dává velmi chybné výsledky i pro množiny se zřetelnými shluky. Pokud v datech existují izolované body (jednoprvkové shluky), mohou dále výrazně zkreslit charakteristiky výsledných shluků.

Metoda **hledá optimální rozklad množiny** (vzhledem k dané kriteriální fci) **na k podmnožin** “tak, aby všechny body měly ke svému středu přibližně stejně daleko”.

Příklad 5.11.

Vhodné využití k-středové metody.

Je dána množina obytných domů svými zeměpisnými souřadnicemi v rozsáhlém prostoru. Firma provádějící služby má v úmyslu zřídit 4 provozovny. Použitím k-means algoritmu se 4 typickými body najde optimální rozložení domů na 4 části, najde středy těchto „shluků" a tam umístí provozovny.

♦

❑ Metody optimalizační k-středové s proměnným počtem shluků

Velmi optimisticky zní zlepšená varianta k-středové metody, která navíc v průběhu iterací testuje, zda okamžité skupiny bodů „mají tendenci" se rozdělit nebo naopak sloučit a tím optimalizuje i původní zadaný počet shluků. První metoda tohoto typu byla pojmenovaná ISODATA. Dodnes se pod tímto názvem objevuje, i když jde obvykle o její zlepšení a automatizované nastavování parametrů, které byly původně zadávány uživatelem. Uvedeme si jednu takovou variantu metody ISODATA

CLASS

hledá optimální počet shluků i nejlepší rozklad;

pro podmínky slučování nebo rozdělování je definováno několik konstant:

THETAN = minimální počet bodů ve skupině, implicitně THETAN = 3

S0 = počáteční rozdělovací práh, implicitně S0 = 0.6

GAMA = maximální počet iterací, implicitně GAMA = 2*m (m je počet objektů)

Algoritmus CLASS

1. Zadání počtu **k** počátečních shluků a **k** počátečních typických bodů.
2. Počáteční rozdělení bodů metodou FORG

Dále je algoritmus iterační, v každém kroku se provádí tato tři rozhodnutí 3-5:

3. **Zmrazení malých shluků**: skupiny o méně než THETAN bodech, které se nezměnily v posledních dvou iteracích, jsou z další analýzy vyloučeny;
4. **Rozdělení shluků**: v m-té iteraci definujeme rozdělovací práh

$$S_m = S_{m-1} + \frac{1 - S_0}{GAMA}$$

Pro každou skupinu se vypočtou nové souřadnice každého bodu jako odchylky od souřadnic těžiště skupiny. Pro j-tou souřadnici se vypočtou průměrné odchylky od těžiště shluku D_{j1} a D_{j2} bodů ležících vpravo a vlevo od těžiště; přitom k1 a k2 je počet těchto bodů vpravo a vlevo :

Podrobněji: V každé iteraci jsou teď nějaké shluky. Pro každý z těchto shluků postupně je spočítáno těžiště T [x, y, …], tedy pro shluk Si je to těžiště Ti [xi, yi, …].

Vezmeme atribut - **souřadnici x,** tedy **xi** tohoto těžiště a postupně všechny body shluku Si, těch je **h** (třeba 10). Pro každý bod (například bod Z [a, b, …]) shluku spočítáme jeho odchylku od těžiště, tedy (xi – a) ... označeno xij. Tyto odchylky sčítáme – zvlášť ty, pro něž ně a>xi = to jsou body ležící vpravo od těžiště (třeba jich je 6 ze všech 10), zvlášť pro ty, kde a<xi ... vlevo (třeba jsou 4 z 10). Každý součet vydělíme počtem bodů vpravo (= k1) a vlevo (= k2) a dostaneme **Dj1 a Dj2**.

$$D_{j1} = 1/k_1 . \sum_{i=1}^{k_1} x_{ij} \qquad D_{j2} = 1/k_2 . \sum_{i=2}^{k_2} x_{ij}$$

Současně se počítá maximum těch odchylek (a-xi) pro body vpravo a vlevo, označíme například max_P a max_L.

Totéž uděláme pro atributy **další y**, …: vždy sčítáme odchylky (yi – b), máme tedy Dj1_x, Dj2_x ... pro souřadnici x a Dj1_y, Dj2_y ... pro souřadnici y atd.

Dále se definují hodnoty

$$a1 = \max_j \frac{D_{j1}}{\max x_{ij}} \qquad a2 = \max_j \frac{D_{j2}}{\max x_{ij}}$$

pro body vlevo a vpravo od těžiště, j = 1, ... ,S

Když máme ty Dj1, ... atd, spočítají se 2 maxima – jedno pro pravé body, jedno pro levé body, ve jmenovateli zlomku jsou max_P a max_L; to se udělá pro všechny souřadnice (pro x, y, …) a z nich se vezme maximum, tedy a1 a a2 je největší z těch pravých (levých) zlomků.

Je-li v m-té iteraci počet skupin menší než 2k (k je počáteční počet shluků), a1 > Sm nebo a2 > Sm a zároveň je počet bodů větší než 2*(THETAN+1), rozdělí se shluk podle té j-té souřadnice, v níž a1 nebo a2 nabývá maximální hodnoty, a to na body vlevo a vpravo od j-té souřadnice těžiště shluku.

5. **Zrušení shluků**: v každém kroku se vypočte minimální vzdálenost skupin

$$TAU = 1/h . \sum_{i=1}^{h} D_i$$

kde h je současný počet skupin, D je minimální vzdálenost těžiště i-té skupiny od těžišť ostatních skupin. Jestliže pro i-tou skupinu platí D_i < TAU a zároveň je počet skupin větší než K/2, zruší se tato skupina a každý její bod se přiřadí ke skupině s nejbližším těžištěm.

Po každé změně se provede FORG. Iterace se provádí do ustálení, maximálně GAMA-krát.

Jak se dá očekávat z použití metody FORG pro každý mezivýsledek, výsledkem jsou **opět kulové „shluky“**.

Uvedený algoritmus však není ideální ani pro optimalizaci rozkladu. Například kritérium pro rozdělování shluků testuje rozdělení jen podél souřadných os (= vždy podle jednoho atributu).

❑ Fuzzy C-means algoritmy

jsou variantou metod optimalizačních se zadaným pevným počtem shluků, ovšem na rozdíl od všech ostatních metod hledají „shluky“ překrývající se. Existují úlohy, kdy takový výsledek dává smysl, i když nejde o typickou shlukovací úlohu (viz příklad níže).

Fuzzy zobecnění minimální odchylky shlukování bylo definováno v roce 1973 [x] a byl vyvinut algoritmus minimalizace kriteriální funkce. Tato práce byla později zobecněna [x] pro nekonečné cílové funkce spolu s algoritmem minimalizace výsledného problému. V roce 1985 [x] byla

formulována modifikovaná verze, která zaručuje konvergenci fuzzy shlukovacího algoritmu (podle c-typických bodů) buď k **lokálnímu minimu** nebo k **sedlovému bodu** cílové funkce.

Shlukovací úloha je definována jako minimalizace výrazu

$$J_m(W, Z) = \sum_{i=1}^{n} \sum_{j=1}^{c} w_{ij}^{m} \| x_i - z_j \|^2_A$$

s podmínkou, že $w_{ij} \in \langle 0,1 \rangle$, $\sum_{j=1}^{c} w_{ij} = 1$ pro $1 \le i \le n$,

$w_{ij} \ge 0$ pro $1 \le i \le n$ a $1 \le j \le c$,

$\sum_{j=1}^{n} w_{ij} > 0$ pro $1 \le i \le c$

kde ***n*** je počet bodů

c je pevný a zadaný počet shluků

m je skalár, tzv. váhový exponent ($m > 1$)

s je dimenze daného prostoru

A je jakákoliv kladná symetrická matice $s \times s$

$x_i \in R^S$, $1 \le i \le n$ jsou dané body v charakteristickém prostoru R^S

$z_j \in R^S$, $1 \le j \le c$ jsou středy shluků

$\| \ \|_A$ je norma skalárního součinu v R^S

w_{ij} představuje míru asociace vzoru (daného bodu) ***i*** se shlukem ***j***

$Z = [z_1, z_2, ... z_c]$ je matice středů shluků $s \times c$

$W = \{ w_{ij} \}$ je matice $n \times c$.

Fuzzy shlukovací algoritmus alternuje mezi počítáním Z a W dokud buď středy Z nebo členská funkce W se neopakují, potom algoritmus končí. Jestliže je m = 1, potom fuzzy shlukovací algoritmus konečně konverguje a zastaví se v lokálním minimu jestliže jako míra vzdálenosti je brán čtverec Euklidovské vzdálenosti.

Matematickým důkazem se zde nebudeme zabývat.

Příklad 5.12.

Jednoduchý příklad využití fuzzy-shlukování je tento: máme dánu množinu 50 ovocných moštů neznámého složení (z jakého ovoce jsou vyrobeny). Dále je dána množina 5 moštů homogenních (jablečný, višňový, pomerančový, grepový, rybízový), ty tvoří pevný počet typických bodů. Otázkou je, ze kterých složek se skládají mošty analyzované.

*Zkoumaná množina moštů je matice **X**. Matice zadaných typických moštů je **Z**. Výsledkem výpočtu je matice **W,** v níž každému zkoumanému moštu je vypočteno 5 hodnot – váha příslušnosti k jednomu z pěti typických moštů. Jinam řečeno, kolik procent kterého moštu zkoumaný vzorek obsahuje.*

Je zřejmé, že součet těchto pěti čísel je roven 1 (=100%).

♦

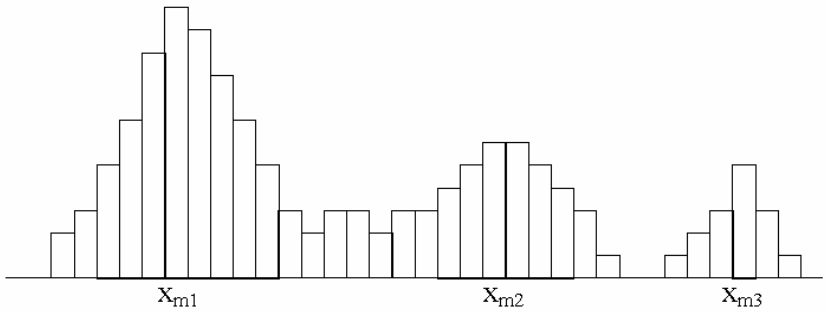
❑ Analýzy modů

Jiný přístup ke shlukování mají metody, hledající nejhustší místa prostoru - módy. **Modus** je hodnota a_m náhodné veličiny (atributu) A, v níž nabývá frekvenční funkce veličiny A lokálního maxima. To znamená, že této hodnoty nabývá maximum bodů. Předpokládá se, že poloha módu odpovídá poloze typického bodu shluku.

Prakticky se variační interval domény atributu rozdělí na třídní intervaly a v nich se sledují četnosti výskytu hodnot. Výsledné relativní četnosti se znázorní histogramem. Z něj je možno odhadovat existenci a polohu módů frekvenční funkce.

Příklad 5.13.

Představme si nejprve jednoatributové objekty s doménou atributu <a, b>. Rozdělíme interval na třídní intervaly, určíme četnosti v těchto třídních intervalech a dostaneme histogram z následujícího obrázku. V něm tvoří módy tři lokální maxima xm1, xm2, xm3. Jsou to hodnoty s (lokálně) nejvyšším počtem bodů, tedy nejhustší, tedy předpokládané „jádro" shluku.



Obrázek 5.9. Jednorozměrné rozložení módů

Pokud jsou módy od sebe odděleny intervaly s nulovou četností (jako mezi módy xm2 a xm3), můžeme je považovat za různé shluky. Pokud ovšem takový přirozený předěl mezi módy neexistuje (jako mezi módy xm1 a xm2), volíme jistou minimální hranici četnosti pro určení předělu mezi shluky.

♦

Objekty ke shlukování jsou obecně popsány n atributy, jde tedy o populaci reprezentovanou n-rozměrným náhodným vektorem

$$\mathbf{A} = (A_1, A_2, \ldots, A_n)$$

Nyní hledáme shluky z existence a polohy módů mnohorozměrné frekvenční funkce. Odhad typu mnohorozměrné frekvenční funkce je zřejmě složitější, než v případě jednorozměrné veličiny. Obecné algoritmy pro její nalezení dosud neexistují.

Odhad polohy módů z frekvenční funkce v n-rozměrném prostoru

1. pro každý z n atributů určíme doménu a tím vymezíme n-rozměrný kvádr D (jako obal množiny bodů)
2. rozdělíme oblast D na n-rozměrné intervaly (reprezentované opět n-rozměrnými kvádry se zvoleným krokem délky hrany ve směru každé souřadné osy)
3. sledujeme četnosti výskytů sledovaných bodů v těchto kvádrech
4. intervaly s největší četnosti, oddělené vzájemně intervaly s malou četností, dávají odhady umístění módů v množině bodů.

N-rozměrný problém nelze znázornit histogramem. I sledování četností se pro větší n stává problémem: rozdělíme-li každý z n variačních intervalů na k úseků, dostaneme k^n n-rozměrných kvádrů. Rozhodnutí o tom, zda 1 bod patří do určitého intervalu, vyžaduje průměrně (pro jeden atribut) k testů, pro n atributů k * n testů. Obdobné množství testů by si vyžádalo porovnání výsledných četností v sousedních intervalech. Celkem by tedy takový logicky jednoduchý algoritmus měl exponenciální složitost.

Proto existuje v literatuře množství metod, odhadujících polohy módů, které jsou založeny na statistických, matematických a jiných přístupech, někdy také s vysokou časovou složitostí a pro větší data nepoužitelných.

Uvedeme si jeden z jednoduchých algoritmů - Kittlerův. Kittler nazývá svou metodu lokálně senzitivní (na rozdíl od metod globálně senzitivních, jako např. ISODATA), protože analyzuje data s ohledem na detaily struktury dat.

Algoritmus ANMO - Kittlerova metoda hledání módů

1. Zadáme α (pro α-souvislé shluky).
2. Ke každému bodu O_i množiny $\mathbf{O} = \{O_1, O_2, \ldots, O_n\}$ sestrojíme jeho α-okolí jako množinu všech bodů O_j takových, že pro něž platí $d(O_i, O_j) < \alpha$.
3. Sestrojíme množinu S takových bodů O_j, které mají ve svém α-okolí alespoň jeden bod rùzný od bodu O_i.
4. Dokud je množina S neprázdná, vybereme z ní libovolný bod L a zaznamenáme počet ostatních bodù v jeho α-okolí, jinak přejdeme k bodu 7.
5. Vybraný bod L vytiskneme i s počtem bodů jeho α-okolí.
6. Vyjmeme bod L z množiny S a zařadíme ostatní body jeho α-okolí do množiny P.
7. Dokud je množina P neprázdná, vybereme z ní (další) bod L s největším α-okolím a přejdeme k bodu 4., jinak přejdeme k bodu 3.
8. Vytiskneme zbývající izolované body (body s prázdným α-okolím).

Výsledkem metody je jedna nebo více posloupností dvojic čísel - pořadové číslo bodu a počet bodů v jeho α-okolí.

Na rozdíl od optimalizačních metod nacházejí analýzy módů přirozené shluky.

Výsledkem je nehierarchický rozklad, kde shluky tvoří seskupení objektů kolem předpokládaných modů, ostatní objekty tvoří izolované body.

Problémem bývá nastavení parametru α znamenajícího "jak vzájemně vzdálené" body ještě patří ke shluku a které již ne. Na tomto parametru závisí výsledný počet a velikost shluků. Obvykle nelze najít optimální rozklad jediným shlukováním.

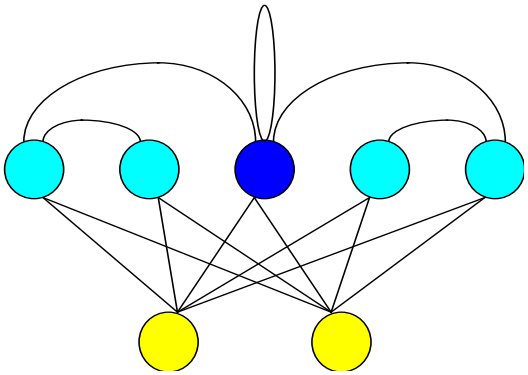
Závěr: Analýzy módů konstruují **přirozené α-shluky pro zadané α**. Pro praxi bývá užitečnější použít vhodně nastavenou hierarchicku metodu se strategií nejbližšího souseda (viz níže), která dává výsledky stejného typu a spolehlivěji najde optimální rozklad.

❑ Kohonenovy mapy

Zcela jiný přístup ke shlukování má jeden z typů neuronových sítí – Kohonenovy mapy. Přiřazují n-dimenzionálním vektorům na vstupu obvykle dvourozměrnou reprezentaci ve výstupní vrstvě neuronů tak, že podobné vektory jsou reprezentovány blízkými si neurony v dané topologii sítě. Na rozdíl od ostatních typů neuronových sítí jde o metodu, která nevyžaduje trénovací množinu, jde o adaptaci bez učitele. Při vhodně nastavených počátečních parametrech sítě je metoda rychlá. Výsledkem je příslušnost objektu ke shluku, charakteristiky se musí dopočítat.

Kompetitivní neuronová síť je tvořena dvěma vrstvami neuronů.

1. Spodní představuje vstupní jednotky, které jsou propojeny se všemi neurony vrstvy výstupní.
2. Vrchní vrstva je tvořena výstupními neurony, které jsou mezi sebou propojeny.



Obrázek 5.10. Kompetitivní síť

Propojení ve výstupní vrstvě je typické **sebeexcitující vazbou** a **inhibičními hranami** vzhledem k ostatním neuronům. Toto propojení vede k posilování neuronu, který byl na počátku nejvíce excitován. Je také **zachována topologie vstupních vektorů**. To znamená, že neurony, které jsou si blízko, rozpoznávají sobě blízké vektory.

Adaptace jednotlivých neuronů spočívá v tom, že při vstupu vstupního vektoru neurony soutěží v tom, který je vstupnímu vektoru nejblíž. Ten neuron který vyhraje, "zahoří". Neuron, který zahořel, sám sebe posiluje samoexcitující vazbou a ostatní potlačuje inhibičními hranami.

Výsledkem je, že neurony, které často vyhrávají pro danou skupinu vzorů se stávají dominantními a ostatní jsou potlačeny. Tedy pro danou skupinu vstupních vektorů zahoří pouze jeden neuron (po naučení sítě).

Výhodou je, že síť se organizuje sama, bez žádného učitele.

Tento způsob je podobný postupu excitace a inhibice v mozku.Každý objekt pak reprezentuje nějaký objekt či třídu objektů ze vstupního prostoru. Tento neuron je pak schopen rozpoznat celou třídu si podobných vstupních vektorů.

Ve skutečném mozku se učení děje postupně tak, že při získávání znalostí se zabírají další a další části mozkové kůry, kde dochází k učení na jednotlivé vzory postupně, tak, jak přicházejí různé druhy vjemů. Výběr oblastí na různé typy vjemů však nejsou náhodné, protože umístění jednotlivých oblastí a základní vlastnosti jsou vrozené.

Výše uvedené principy samoorganizace a adaptace jsou aplikovány v Kohonenových mapách.

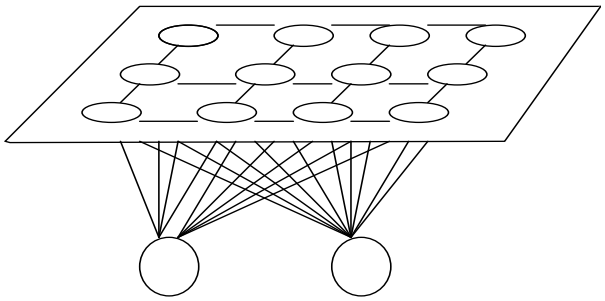
Hlavní ideou těchto neuronových sítí je nalézt prostorovou reprezentaci složitých datových struktur. Tedy aby třídy si podobných vektorů byly reprezentovány neurony blízkými si v dané topologii.

Tato vlastnost je typická i pro skutečný mozek, kde např. jeden konec sluchové části mozkové kůry reaguje na nízké frekvence, zatímco druhý konec reaguje na frekvence vysoké.

Tímto způsobem lze zobrazit mnohodimenzionální prostor do prostoru jednoduššího, nejčastěji dvoudimenzionálního.

Díky zkušenostem bylo zjištěno, že při vstupu zahoří vítězný neuron, ať už s nebo bez samoexcitujících a inhibičních hran. Důsledkem toho je, že tyto hrany není nutno implementovat.

Topologie může být například následující



Obrázek 5.11. Kohonenova mapa

Algoritmus učení Kohonenovy mapy

1. **Inicializace sítě**

   Definuj $w_{i,j}(t)(0 \le i \le n-1)$ jako váhu mezi vstupem i a neuronem j v čase t.

   Inicializuj tyto váhy malými náhodnými čísly. Nastav počáteční okolí kolem neuronu j, $N_j(0)$ na maximum (obvykle tak, aby toto okolí pokrývalo celou výstupní vrstvu).

2. **Předložení vstupního vektoru**

   Předlož vstup ve formě $x_0(t), x_1(t), \ldots, x_{n-1}(t)$, kde $x_i(t)$ je vstup uzlu i v čase t.

3. **Výpočet vzdáleností (stanovení vítěze kompetice)**

   Vypočti vzdálenosti d_j mezi vstupním vektorem a každým neuronem následovně :

$$d_j = \sum_{i=0}^{n-1} (x_i(t) - w_{i,j}(t))^2$$

4. **Výběr minimální vzdálenosti**

   Určení vítězného neuronu prostřednictvím výše vypočteného d pro každý neuron. Vítězný neuron označme jako j^*.

5. **Úprava vah**

   Úprava vah pro neuron j^* a jeho sousedy definované jeho okolím. Nové váhy jsou dány vztahem:

   $w_{i,j}(t+1) = w_{i,j}(t) + \alpha(t)\left(x_i(t) - w_{i,j}(t)\right)$, pro j patřící do okolí vítězného neuronu $j^*, 0 \le i \le n-1$, kde $0 \prec \alpha(t) \prec 1$, přičemž hodnoty $\alpha(t)$ a velikost okolí neuronu se postupně snižují v čase s cílem stabilizovat nastavené váhy a lokalizovat místa maximální aktivity.

6. **Opakování od bodu 2.**

Proces shlukování lze vysvětlit prostřednictvím **funkce hustoty pravděpodobnosti**. Tato funkce reprezentuje statistický nástroj popisující rozložení dat v prostoru. Pro daný bod prostoru lze tedy stanovit pravděpodobnost, že vektor bude v daném prostoru nalezen. Je-li dán vstupní prostor a funkce hustoty pravděpodobnosti, pak je možné dosáhnout takové organizace mapy, která se této funkci přibližuje (za předpokladu, že je k dispozici reprezentativní vzorek dat). Jinými slovy řečeno, jestli jsou vzory ve vstupním prostoru rozloženy podle nějaké distribuční funkce, budou v prostoru vah rozloženy vektory analogicky.

Poznámka:

Pokud chceme testovat účinky jednotlivých parametrů, použitých v algoritmu, pak je vhodné vyzkoušet změnu všech tří:

1. **Koeficient učení,** poslední zadávaná položka. Jedná se o počáteční koeficient, který určuje jak moc se vítězný neuron a jeho okolí ztotožní se vstupním vektorem. Tento koeficient se postupem času snižuje na nulu, takže nakonec nemá žádný vliv. Nemá smysl zadávat tento koeficient větší než 1, protože účinný bude od 1 níže.
2. **Okolí** - počáteční okolí vítězného neuronu, ve kterém budou upravovány váhy. Tato položka se s časem opět snižuje. Je vhodné ji na počátku zvolit tak, aby první okolí pokrývalo celou plochu mapy.
3. **Počet neuronů na hranu** - rozměr čtverce výstupní mapy, kde se jedná o mapu s (počet * počet) neurony. Není vhodné volit tento počet příliš vysoký, neboť s růstem neuronů roste také výpočetní doba.

❑ Gravitační shlukovací metoda

Jen jako ukázku toho, že uvedené typy metod nejsou všechny, které se v literatuře objevují a že některé spočívají na zcela odlišných principech, si uvedeme tzv. gravitační metodu, která shlukuje na základě gravitačního zákona

$$F = \kappa * m_1 * m_2 / r^2$$

kde κ je gravitační konstanta, zadávaná pro výpočet analytikem, (0.0001)

m jsou hmotnosti přitahovaných těles, hmotnost tělesa je rovna součtu hmotností objektů,

hmotnost objektu je 1

r je vzdálenost mezi tělesy

F je výsledná působící síla.

Dále jsou zadány parametry

R_{min}, udávající minimální vzdálenost, při které se objekty spojí a

P udávající, kdy se posun neuskuskuteční, P= $R_{min}/100$

Gravitační algoritmus shlukování:

V každém kroku se

1. spojí objekty se vzdáleností menší než R_{min},
2. vypočítají se posuny všech objektů vůči ostatním
3. objekty posunou.

kroky se opakují tak dlouho, až je matice posunů nulová.

Shrnutí pojmů 5.2.

Metody nehierarchické a jejich typy.

Metody optimalizační.

Problém počátečního rozkladu, typické body. Metody pro definování typických bodů počátečního rozkladu.

Náhodný výběr typických bodů, zadání typických bodů, řízený výpočet typických bodů podle rozložení množiny shlukovaných bodů.

Metody optimalizační k-středové s pevným počtem shluků a jejich varianty. Typ výsledku shlukování.

Metody optimalizační k-středové s proměnným počtem shluků. Typ výsledku shlukování.

Fuzzy C-means algoritmy a úlohy touto metodou řešené.

Analýzy modů, algoritmus Kittlerův. Typ výsledku shlukování.

Shlukování pomocí Kohonenovy mapy.

Otázky 5.2.

1. Které typy metod nehierarchických shlukovacích znáte?
2. Proč neexistuje jediná spolehlivá shlukovací metoda?
3. Čím jsou charakteristické shlukovací metody optimalizační, k-středové?
4. Jakými způsoby se určují počáteční typické body pro shlukování a jejich počet?
5. Uveďte základní algoritmus shlukovací metody k-středové s pevným počtem shluků.
6. Uveďte základní body algoritmu shlukovací metody k-středové s proměnným počtem shluků.
7. Uveďte princip fuzzy k-středového shlukování.
8. Uveďte princip shlukování pomocí metod analýzy módů.
9. Uveďte princip shlukování pomocí neuronových sítí.
10. Podle čeho se budete rozhodovat při výběru metody pro shlukování konkrétní množiny objektů?
11. Existují jiné metody shlukovací, než uvedené typy? Proč?

Úlohy k řešení 5.2.

1. Pro data z kapitoly 5.1./1 BANKA navrhněte, kterou metodou budete data shlukovat, případně navrhněte pro zvolenou metodu vhodné parametry.

2. Totéž proveďte pro data 5.1./2 GYMNÁZIUM.

3. Najděte alespoň 3 praktické úlohy, kdy je vhodné použít optimalizační k-středový algoritmus shlukování a zdůvodněte, proč.

5.3. Shlukování hierarchické

Cíl Po prostudování této kapitoly budete umět

- popsat princip hierarchického shlukování aglomerativního i divizivního,
- zdůvodnit, proč není vhodné používat metody shlukující jen dvojice shluků,
- rozhodnout, kdy je k řešení úlohy vhodné najít hierarchii shluků,
- určit pro úlohu vhodnou míru podobnosti a vhodnou shlukovací strategii,
- pro praktické problémy najít optimální shlukovací hladiny a interpretovat výsledek.

Výklad

❑ Shlukování hierarchické

Hierarchické shlukovací metody hledají nejen prostý rozklad objektů na shluky, ale jako výsledek dávají celou hierarchii takových rozkladů. V dané množině objektů se může vyskytovat řada malých shluků, které opět mohou být vzájemně různě vzdálené. Tak mohou tvořit „shluky shluků“ a vytvářet méně shluků větších (viz například testovací data 3). Takových hierarchických stupňů může existovat několik.

Postupně byla vyvinuta různými autory skupina metod, vytvářejících hierarchii rozkladů daných **n** bodů. Základní dva přístupy jsou

- **divizivním,** shora dolů, rozdělující postupně celou množinu bodů vždy na dva shluky, až dojdou k jednotlivým bodům,
- **aglomerativním**, zdola nahoru, vycházejících od jednotlivých bodů jako jednobodových shluků a shlukujícím postupně vždy nejbližší shluky, až dojdou k jedinému shluku ze všech bodů.

Nevýhodou obou přístupů může být příliš velký počet rozkladů. Prakticky se totiž využívá většinou jen několik málo stupňů rozkladu. Existují však takové modifikace hierarchického shlukování, které umějí počet stupňů rozkladu významně snížit.

❑ Shlukování aglomerativní základní = po dvojicích shluků

Je dána množina objektů **O** a koeficient vzdálenosti shluků **V**.

Princip algoritmu procedury aglomerativní

1. Počáteční rozklad tvoří jednoobjektové shluky {Oi}; zvolíme míru vzdálenosti objektů, vypočte se **matice vzdáleností** objektů.
2. Nalezne se nejmenší **vzdálenosti shluků** (tzv. shlukovací hladina hierarchie).
3. Spojením těchto nejbližších shluků do společného shluku se vytvoří vyšší stupeň hierarchie, ostatní shluky zůstanou nezměněny.
4. Vypočtou se charakteristiky shluků aktuální hladiny rozkladu.
5. Pokud existuje více než 1 shluk, opakuje se od bodu 2.

V algoritmu jsme použili dva nové pojmy, které nejprve vysvětlíme.

❑ Matice nepodobností - vzdáleností objektů

Na počátku algoritmu aglomerativního se chápe každý objekt jako samostatný shluk. Vzdálenost těchto jednoobjektových shluků je rovna vzdálenosti objektů. Metriky pro výpočet vzdáleností jsme definovali v úvodu kapitoly 5. Zvolíme tedy míru vzdálenosti pro náš výpočet (*například Euklidovskou vzdálenost*).

Pak maticí vzdáleností rozumíme čtvercovou matici se všemi vzájemnými vzdálenostmi objektů. Protože je to matice symetrická, stačí počítat jen její polovinu.

Příklad 5.15.

Je dáno 7 objektů – lidí, spočítaná matice vzdáleností je:

	O1	O2	O3	O4	O5	O6	O7
O1	0	100	100	50	33	25	20
O2		0	100	50	50	33	33
O3			0	100	33	25	20
O4				0	33	20	25
O5					0	100	100
O6						0	100
O7							0

♦

❑ Míra vzdálenosti shluků – shlukovací strategie

Další nový pojem v algoritmu je vzdálenost shluků. Když se v průběhu shlukování vytvoří shluky několika objektů a opět se hledají nejbližší shluky, je nutné tento vztah shluků definovat. V literatuře se za řadu let objevily různé míry pro vzdálenost shluků. Někdy se jim říká shlukovací strategie.

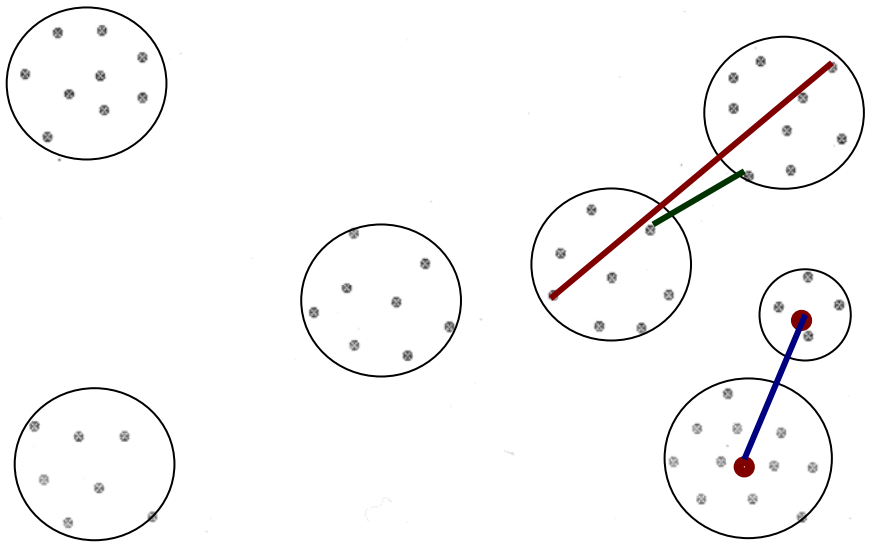
Nejznámější používané strategie jsou

- strategie nejbližšího souseda
- nejvzdálenějšího souseda,
- průměrné vzdálenosti objektů
- mediánová
- centroidní
- Ward-Wishartova

Přesné vztahy pro jejich výpočet uvedeme níže. Protože je potřeba znát opět vzdálenosti všech shluků navzájem, jsou opět uspořádány do matice vzdáleností.

Příklad 5.16.

Testovací data 1 jsou v průběhu shlukování ve stavu, kdy již existuje 6 shluků podle následujícího obrázku. Vzdálenost shluků strategií nejbližšího souseda (zeleně), nejvzdálenějšího souseda (červeně) nebo centroidní (modře) je zobrazena na sousedních shlucích. Obdobně by se určila mezi všemi shluky navzájem.



♦

Některé z uvedených strategií bývají součástí SW balíků pro dolování. Dávají různé typy výsledků – shluky přirozené i shluky kulové.

Výhoda metody je v rekurzivitě algoritmu - není nutné počítat vzdálenosti shluků z původních souřadnic bodů, ale v každém kroku se pouze přepočítají vzdálenosti nově vzniklých shluků od ostatních. K tomu však je potřeba definice vzdálenosti shluků.

Ukazuje se, že všechny strategie se dají formulovat do jediného obecného schématu s několika parametry αi, αj, β, γ. Strategie se liší volbou těchto parametrů.

Obecné schéma pro výpočet vzdálenosti shluku U od shluku $R = P \cup Q$ je

$$V(U, R) = \alpha i \, .V\,(U, P) + \alpha j \, .V\,(U, Q) + \beta . V\,(P, Q) + \gamma \, . \, |V\,(U, P) - V\,(U, Q)|$$

Protože o něco níže ukážeme, že tento základní algoritmus má nedostatky, které jeho výsledky někdy zcela znehodnotí, nebudeme přesné vztahy pro jednotlivé strategie zatím uvádět. Uvedeme je až po zobecnění metody.

❑ Dendrogram

Problémem je, že všechny strategie produkují velké množství hierarchických stupňů a až „ručním“ výběrem analytik vybírá jednu nebo několik výsledných.

Výsledek hierarchie je vhodné vykreslit do grafu – shlukovacího stromu nazývaného **dendrogram.** Ovšem i dendrogram je jen jakýsi „polotovar“ k vyhodnocení. Podívejme se podrobněji, proč.

U klasické aglomerativní metody se v každém kroku – v každé hladině shlukují vždy 2 nejbližší shluky. Tedy pro **m** objektů existuje m-1 shlukovacích hladin. Je zřejmé, že tolik rozkladů není k ničemu užitečných. Je potřeba vybrat jen několik „nejlepších“. K tomu se využívá dendrogram s hodnotami shlukovacích hladin. Vyhodnocení provádí buď analytik, nebo je ho možno automatizovat. Z dendrogramu se vyhledávají

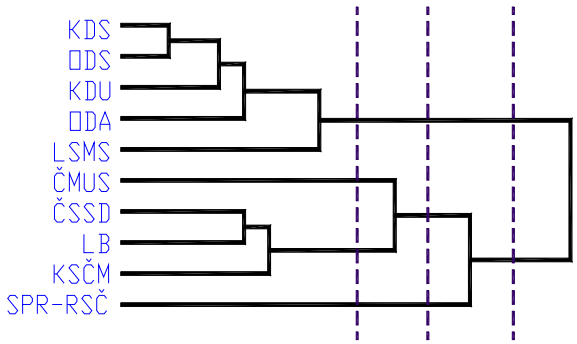
- největší rozdíly mezi sousedními hladinami … znamenají nejlepší hierarchické stupně
- hladina se zadaným počtem shluků … pokud existuje důvod nebo známe počet shluků předem
- zadání velikosti hladiny … pokud zadáme vzdálenost mezi shluky.

Obvykle nevyžadujeme příliš mnoho shluků, proto se dendrogram analyzuje „od konce“. Požadujeme-li kompletní „rozumnou“ hierarchii, najdeme několik málo (například 2 až 5) největších rozdílů mezi hladinami a rozklady na těchto stupních považujeme za výslednou hierarchii.

Příklad 5.17.

V letech 1995-96 bylo v parlamentu ČR celkem 10 stran a jejich 2049 hlasování bylo shluknuto následovně – pro zobrazení je použit dendrogram. Největší rozdíl mezi hladinami je mezi předposlední a poslední. To znamená, že nejlepší rozklad je na 2 shluky (vidíme, že odpovídají klasickému dělení politických stran na pravici a levici, s jedinou „výjimkou“ SRP-RSČ). Rozklad o hladinu níže tuto výjimku oddělí do samostatného jednoobjektového shluku.

Čteme-li dendrogram zprava, skutečně strany si nejbližší se později spojily.



♦

❑ Charakteristiky výsledných shluků

Dendrogram nám dává jistou informaci o rozložení objektů do shluků, ale necharakterizuje výsledné shluky. V předcházející kapitole jsme si uváděli, že charakterizovat shluky přirozené nelze jen průměrnými hodnotami atributů (těžištěm shluku). Je těžké najít stručný popis výsledných shluků. Pokusíme se o to následujícím způsobem. Do výsledku uvedeme:

Název dat, rozměry dat (počet objektů, počet atributů), názvy atributů.
Zvolená metoda a její parametry.

Globální charakteristiky dat – pro každý atribut jeho min, max, avg, std.

min:	0.0	0.0	11.0	0.4
max:	9.0	18.0	58.0	23.6
avg:	2.3	5.2	23.8	12.0
std:			...	

Počet výsledných shluků (případně v každé shlukovací hladině).

Pro každý shluk (případně v každé shlukovací hladině):

Shluk 1
počet objektů: 20
objekty: O1, O4, O8, ...
součet čtverců chyb: 30.8
průměrná vzdálenost bodu od těžiště: 21.5

avg:	2.3	8.2	33.8	12.0
std:			...	
min:		2.3	8.0	21.2	0.4
max:	2.3	8.3	34.8	21.9

Při vyhodnocování výsledků pak sledujeme, které hodnoty atributů jsou uvnitř shluku málo rozptýlené. Tyto atributy pak jsou charakteristické pro shluk. Které atributy nabývají hodnot rozptýlených, nebudeme pro charakteristiku shluku užívat.

Příklad 5.18.

Jsou dána data o 156 pacientech s těmito 9 atributy:

pohlaví	[0=muž, 1=žena]
věk	[0–82]
stav	[0=svob, 1=ženatý/vdaná, 2=rozved, 3=vdov]
měsíc_přijetí	[1-12]
puls	
tlak	
zlozvyk	[ne_alkoh+nekouří, ne_alkoh+kouří, alk_málo+kouří, alk_hodně+kouří]
diagnóza	0=post_páteře, 1=nemoc_srdce, 2=cukrovka, 3=intoxikace, 4=cirhóza_jater, 5=rakovina, 6=epilepsie, 7=kolapsy_slabost, 8=chudokrevnost]
výsledek	[0=domů, 1=přeložen_jinam, 2=zemřel]

Z výsledných 5 shluků byly vyhodnoceny charakteristiky a z nich jsou provedeny tyto závěry (tučně jsou charakteristické hodnoty atributu pro shluk, tence netypické, uvedené jen pro úplnost):

Shluk 1: pohl=0.4, **věk=70**, **stav= vdov**, měs=5, zloz=4, puls=70, tlak=170, **diag=rak, výsl=zemřel**

Shluk 2: **pohl=0.14**, věk=48, **stav= {svob,rozv}**, měs=6, **zloz=alkohol, puls=72, tlak=181, diag=epil, výsl=jinam**

Shluk 3: **pohl=0.91**, věk=58, stav= {vdaná}, **měs=11**, zloz=, puls=68, tlak=140, **diag=ledviny**, výsl=domů

Shluk 4: pohl=0.22, věk=45, stav= {ženatý,vdov}, měs=2, zloz=alkoh, puls=70, tlak=170, diag=rak, výsl=zemřel

Shluk 5: pohl=0.62, věk=59, stav= rozv, měs=7, zloz={nekouří,málo_alk}, puls=75, tlak=172, diag=rak, výsl=domů

Interpretace výsledků:

První shluk jsou starší pacienti (muži i ženy), ovdovělí s rakovinou, zemřeli.

Druhý shluk tvoří převážně muži, žijící osaměle, holdující alkoholu, s epilepsií.

Zajímavý třetí shluk tvoří starší ženy, které na podzim mají potíže s ledvinami.

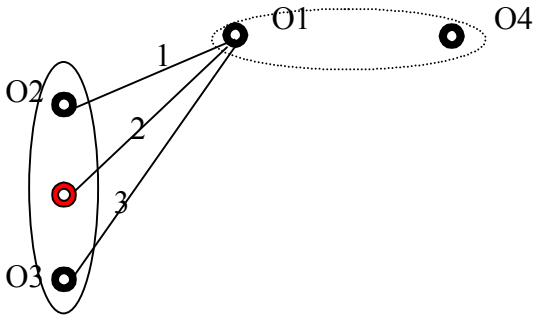
atd.

♦

❑ Shlukování aglomerativní definitní = po n-ticích shluků

Na tomto místě je nutné zdůraznit důležitý a ne obecně známý fakt. Původní hierarchické metody byly formulovány tak, že se v každém kroku shluknou **dva** nejbližší shluky vzhledem ke zvolené strategii shlukování. V případě, že není použit nejbližší soused a v některém hierarchickém kroku má stejnou nejmenší vzdálenost více dvojic či n-tic shluků, může dojít ke značné chybě výsledku.

Podívejme se podrobněji, proč. Na následujícím obrázku x. jsou tři černé body O1, O2, O3. Platí, že vzdálenosti V(O1, O2) = V(O2, O3). Shlukovací algoritmus, který bere vždy dva nejbližší shluky, si vybere body O2,O3 k vytvoření shluku. Potom se přepočítává vzdálenost tohoto nově vzniklého shluku od všech ostatních shluků. Na obrázku jsou úsečkami zobrazeny tři vzdálenosti podle strategie nejbližšího (1), nejvzdálenějšího (3) souseda a mediánové (2). Je zřejmé, že vzdálenosti 2 a 3 jsou větší, než 1. Pokud v následujícím kroku shlukování (mediánovou strategií) bude existovat nejmenší vzdálenost menší, než 2, dojde ke vzniku nového shluku podle ní. Může se tak stát, že bod se dostane do shluku jiného, než ke svému nejbližšímu sousedovi.



Obrázek 5.13. Vzdálenost shluků strategií nejbližšího (1), nejvzdálenějšího (3) souseda a mediánovou (2).

Toto shlukování je tak závislé na pořadí objektů a může při různém pořadí čtení objektů dávat rozdílné výsledky.

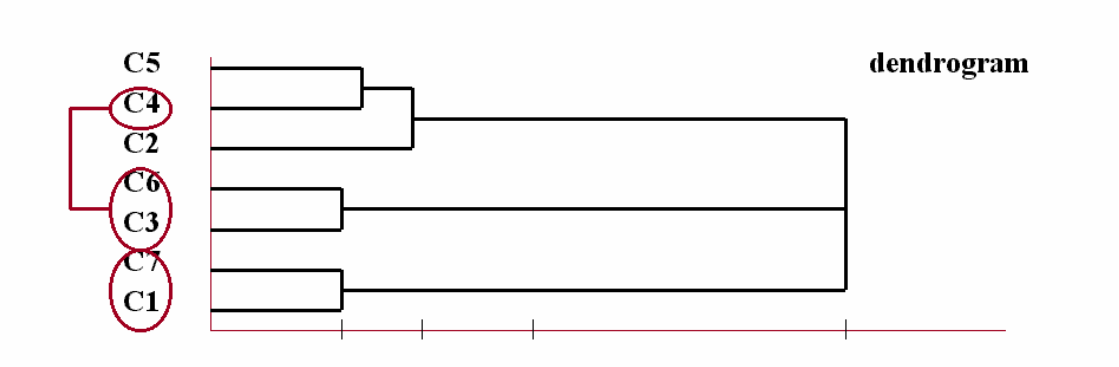
Některé SW systémy dosud používají chybný základní algoritmus a proto mohou dávat naprosto chybné výsledky!!!

Příklad 5.19.

Jsou dána data o 7 lidech. Ve vypočtené matici vzdáleností je nejmenší vzdálenost v 1. kroku shlukování rovna 20. Tato stejná nejmenší hodnota je mezi objekty O1-O7, O3-O6 a O6-O4.

	O1	O2	O3	O4	O5	O6	O7	
O1	**0**	**100**	**100**	**50**	**33**	**25**	**20**	**O1–O7-O3**
O2		**0**	**100**	**50**	**50**	**33**	**33**	
O3			**0**	**100**	**33**	**25**	**20**	
O4				**0**	**33**	**20**	**25**	**O6-O4**
O5					**0**	**100**	**100**	
O6						**0**	**100**	
O7							**0**	

Pro shlukování byl použit nejvzdálenější soused, výsledný dendrogram je

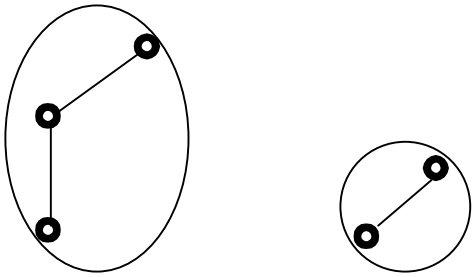


Obrázek 5.14. Dendrogram

Vidíme, že objekt O4, který má obecně nejblíže k O6, se díky zvolené strategii dostal dokonce do úplně jiného shluku! Přitom při jiném uspořádání objektů bychom mohli dostat jiný výsledek, protože jako první pro shlukování by se mohla vzít jiná dvojice bodů.

♦

Základní algoritmus byl později [11] zobecněn tak, že se v každém kroku shlukování spojují **všechny** shluky, které mají stejnou vzájemnou minimální vzdálenost. Přitom v jednom kroku nejen může vzniknout více shluků, ale může vzniknout nový shluk z více předcházejících – viz obrázek.



Obrázek 5.15. Stejné nejmenší vzdálenosti shluků

❑ Shlukovací strategie

Nyní si uvedeme vztahy pro současné shlukování všech n-tic se stejnou hodnotou minimální vzdálenosti:

Označíme:

$\mathbf{A} = \{A\}, \quad \mathbf{B} = \{B\}$	. . . jednoprvkové shluky
$\mathbf{U}, \mathbf{P}_1, \ldots, \mathbf{P}_t, \mathbf{Q}_1, \ldots, \mathbf{Q}_s$	. . . libovolné shluky
$\mathbf{P} = \mathbf{P}_1 \cup \ldots \cup \mathbf{P}_t, \quad \mathbf{Q} = \mathbf{Q}_1 \cup \ldots \cup \mathbf{Q}_s$	
$\lvert \mathbf{P} \rvert, \lvert \mathbf{Q} \rvert, \lvert \mathbf{U} \rvert$	. . . počty objektù shlukù
$D_{\_\_}$	. . . koeficient vzdálenosti shlukù

Pro všechny strategie platí

$V(A,B) = d(A,B)$

Strategie nejbližšího souseda

$$V(U,P) = \min_i \{ d(U,P_i) \}$$

$$V(U,P) = \min_i \left\{ \min_j \{ d(P_i,Q_j) \} \right\}$$

Strategie nejvzdálenějšího souseda

$$V(U,P) = \max_i \{ d(U,P_i) \}$$

$$V(U,P) = \max_i \left\{ \max_j \{ d(P_i,Q_j) \} \right\}$$

Strategie Ward-Wishartova

$$V(U,P_1 \cup P_2) = \frac{(|U|+|P_1|)\cdot V(U,P_1)+(|U|+|P_2|)\cdot V(U,P_2)-|U|\cdot V(P_1,P_2)}{|P_1 \cup P_2|+|U|}$$

$$V(U,P) = \frac{1}{|U|+|P|}\cdot\left(\sum_{i=1}^{t}(|U|+|P_i|)\cdot V(U,P_i)-\frac{|U|}{|P|}\cdot\sum_{i,j}^{\binom{t}{2}}(|P_i|+|P_j|)\cdot V(P_i,P_j)\right)$$

$V(P,Q) =$

$$= \frac{1}{|P|+|Q|}\cdot\left(\sum_{i,j}^{t\times s}(|P_i|+|Q_j|)\cdot V(P_i,Q_j)-\frac{|Q|}{|P|}\cdot\sum_{i,j}^{\binom{t}{2}}(|P_i|+|P_j|)\cdot V(P_i,P_j)-\frac{|P|}{|Q|}\cdot\sum_{i,j}^{\binom{s}{2}}(|Q_i|+|Q_j|)\cdot V(Q_i,Q_j)\right)$$

Strategie centroidní

$$V(U,P) = \frac{1}{|P|}\cdot\sum_{i=1}^{t}|P_i|\cdot V(U,P_i)-\frac{1}{|P|^2}\cdot\sum_{i,j}^{\binom{t}{2}}|P_i|\cdot|P_j|\cdot V(P_i,P_j)$$

$V(P,Q) =$

$$= \frac{1}{|P|\cdot|Q|}\cdot\left(\sum_{i,j}^{t\times s}(|P_i|+|Q_j|)\cdot V(P_i,Q_j)-\frac{1}{|P|^2}\cdot\sum_{i,j}^{\binom{t}{2}}(|P_i|+|P_j|)\cdot V(P_i,P_j)-\frac{1}{|Q|^2}\cdot\sum_{i,j}^{\binom{s}{2}}(|Q_i|+|Q_j|)\cdot V(Q_i,Q_j)\right)$$

Strategie průměrné vzdálenosti objektů

$$V(U,P) = \frac{\sum_{i=1}^{t}|P_i|\cdot V(U,P_i)}{|P|}$$

$$V(P,Q) = \frac{\sum_{i,j}^{t\times s} |P_i| \cdot |Q_j| \cdot V(P_i, Q_j)}{|P| \cdot |Q|}$$

Strategie mediánová

$$V(U,P) = \frac{\sum_{i=1}^{t} V(U, P_i)}{t} - \frac{\sum_{i,j}^{\binom{t}{2}} V(P_i, P_j)}{t^2}$$

$$V(P,Q) = \frac{\sum_{i,j}^{t\times s} \cdot V(P_i, Q_j)}{t \cdot s} - \frac{\sum_{i,j}^{\binom{t}{2}} V(P_i, P_j)}{t^2} - \frac{\sum_{i,j}^{\binom{s}{2}} V(Q_i, Q_j)}{s^2}$$

Tak může být sestaven jediný obecný algoritmus aglomerativních metod. Jednotlivé metody se v něm liší pouze výpočtem vzdáleností shluků.

❑ Algoritmus aglomerativní definitní = po n-ticích

Pro množinu objektů **O** se sestrojí posloupnost rozkladů Ωo, Ω1, . . ., Ωn-1, v ní se přiřadí každému shluku S reálné nezáporné číslo h (S) nazývané **shlukovací hladinou**, odpovídající jeho vzniku, takto:

1. Počáteční rozklad Ωo tvoří jednotlivé objekty = jednoprvkové shluky, h(S)=0
2. Rozklad Ωi = {Si1, Si2, ..., Sik} je rozklad v i-tém kroku procedury, v něm jsou shlukům Sij přiřazena čísla h (Sij) a

$$\mu i = \min_{x\neq y,\ x,y=1,..,k} \mathbf{V}(Six, Siy)$$

je minimální hodnota koeficientu vzdálenosti V. Potom každý μi-souvislý shluk shluků Sik_1, Sik_2, ..., Sik_v rozkladu Ωi přechází do následujícího rozkladu

$$\Omega i+1 = \{Si+1,1, \ldots, Si+1,k\}$$

jako shluk $Si+1,l = Si+1,1 \cup \ldots \cup Si+1,k$

přičemž definujeme h(Si+1,l) = μi. Ostatní shluky zůstávají v novém rozkladu nezměněny.

3. V posledním kroku je Ωn = {Sn1}={**O**} o jediném shluku = celé množině objektů,

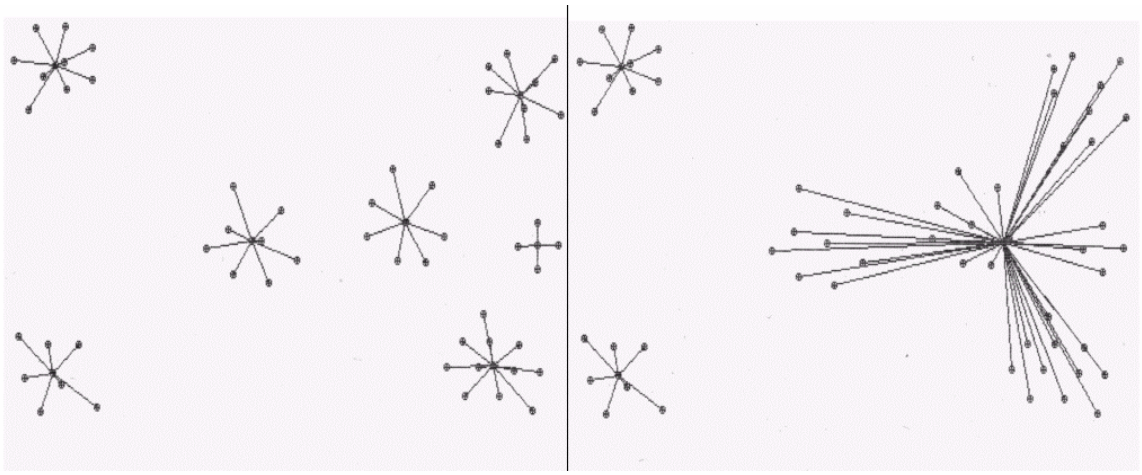
$$h(Sn1) = \mu\ n-1, \text{ kde } \{Sn1\}$$

Výhodou tohoto algoritmu pro všechny strategie je především odstranění zmíněné možné chyby výsledného rozkladu, popsané u základního algoritmu, shlukujícího po dvojicích.

Další výhodou je menší počet shlukovačích hladin, tedy menší počet hierarchických úrovní. I v tomto případě jich však bývá často mnohem více, než je reálně potřebné. Výrazně se zrychlí i zkvalitní výsledek u dat s pravidelnější strukturou (*viz například data 6 – homogenní, která se shluknou v jediném kroku*).

Příklad 5.20.

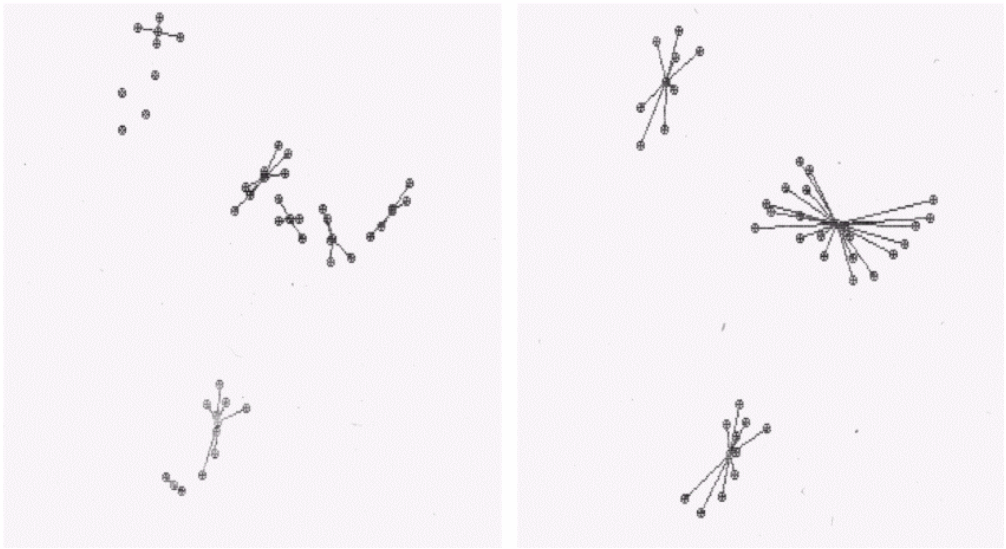
Nejbližší soused, 2 nejlepší rozklady



♦

Příklad 5.21.

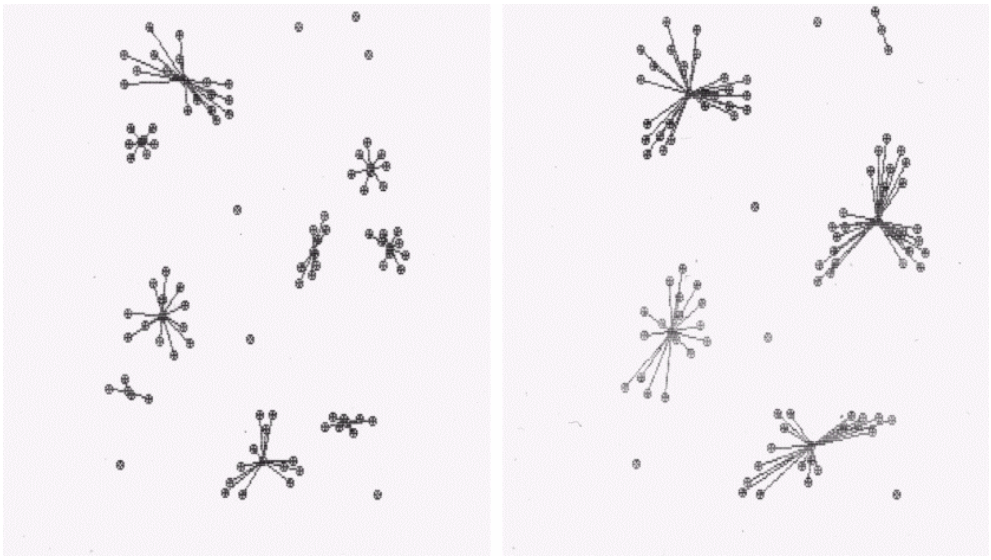
Nejbližší soused, 2 nejlepší rozklady



♦

Příklad 5.22.

Nejbližší soused, 2 nejlepší rozklady, zřetelné jsou i izolované body

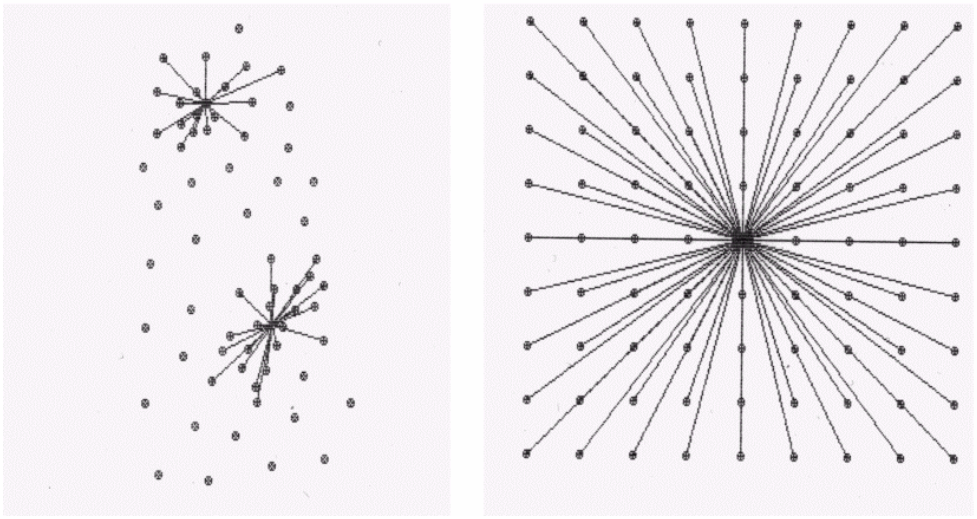


♦

Příklad 5.23.

Nejbližší soused na datech bez shluků, jen s hustšími místy.

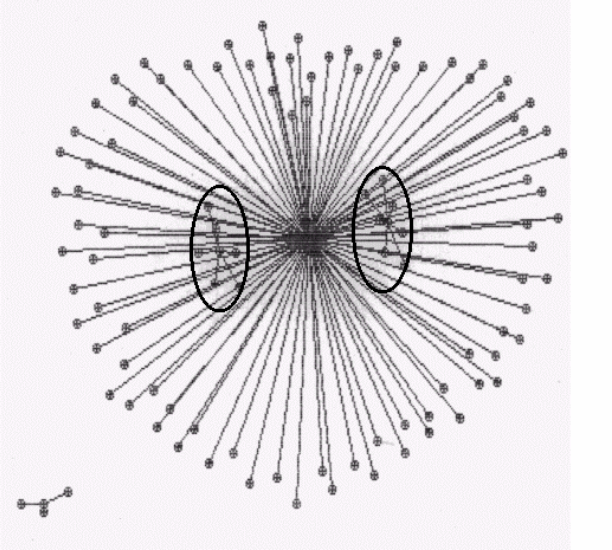
Nejbližší soused na homogenních datech – výsledek s jedinou shlukovací hladinou.



♦

Příklad 5.24.

Nejbližší soused, data odolávající mnoha algoritmům. Nejlepší hladina jasně oddělí od sebe jak 3 malé shluky vně i uvnitř, ale najde i shluk v uzavřeném tvaru srdce.



♦

❑ Tolerance při aglomerativním shlukování

Posledním příspěvkem ke zkvalitnění výpočtu je možnost využití zadané tolerance při určování nejmenší vzdálenosti shluků. Při velkém počtu shlukovaných objektů může být stále počet hierarchických stupňů vysoký. Přitom může docházet k tomu, že se vzájemné vzdálenosti od sebe liší jen málo a tento rozdíl nehraje pro rozklad žádnou podstatnou roli.

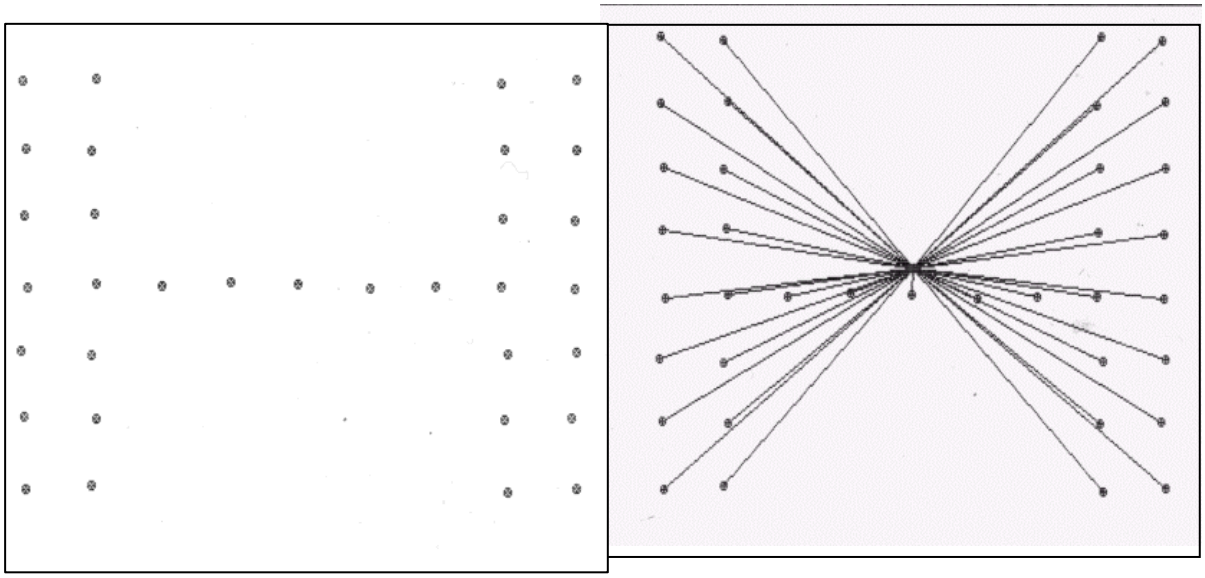
Pokud tedy používáme definitní aglomerativní algoritmus, můžeme urychlit shlukování takto: definujeme toleranci (například v procentech), o kolik se mohou lišit vzdálenosti shluků od nejmenší vzdálenosti, aby byly pořád ještě považovány za „stejné“. Tolerance může být zadána analytikem, nebo nastavována automaticky. Tak se na jedné hladině shlukne současně více shluků, počet hladin se zmenší a výpočet se zkrátí.

Použití tolerance je zvláště výhodné u experimentálních dat, kde se již při měření vyskytuje jistá chyba a přesný výpočet vzdáleností je dokonce zkreslující.

Příklad 5.25.

Pro otestování schopností shlukovacích metod byla vytvořena umělá data, tzv. H-data, kde jednotlivé body jsou zatíženy chybou ± 0.1. Až na tuto chybu data tvoří jeden homogenní přirozený shluk.

Použitím metody Ward-Wishartovy s tolerancí 20 % došlo k výpočtu jediné hladiny:

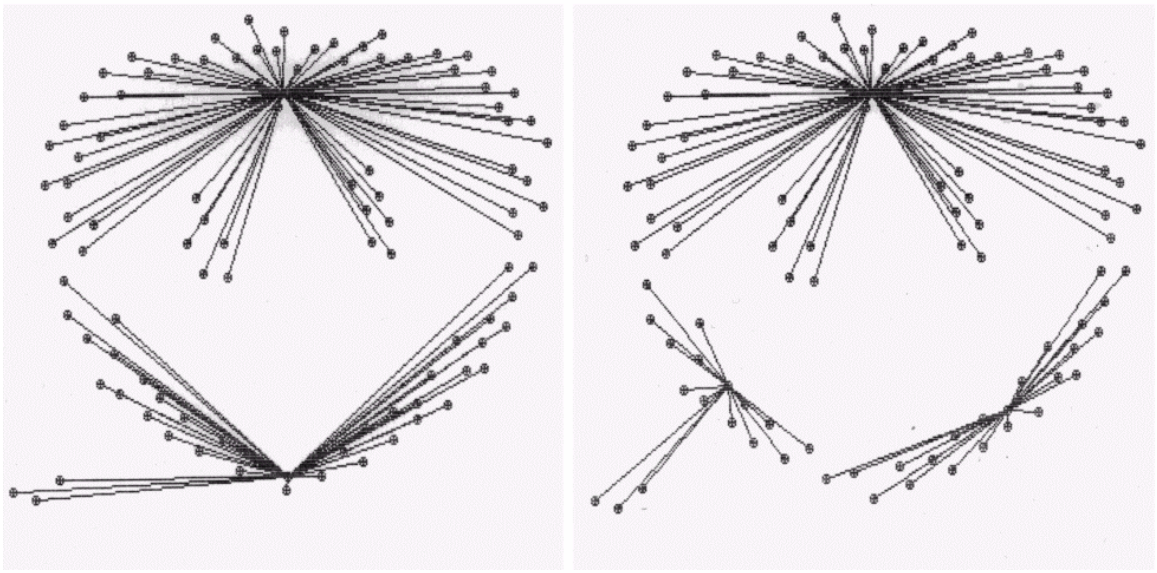


♦

Příklad 5.26.

Data srdce, metoda Ward-Wishartova, tolerance 10%.

Přesto, že metoda byla vyvinuta pro nalezení přirozených shluků a při vhodné toleranci na „stejné“ nejmenší vzdálenosti vykazuje dobré výsledky, tato data nezvládne. Výsledek má obdobný efekt, jako metody k-středové.



♦

❑ Shlukování divizivní

Divizivní hierarchické metody vycházejí z jednoho shluku, vytvořeného celou množinou objektů. Postupným rozdělováním dojdou až k jednoprvkovým shlukům. Vzájemně se liší způsobem

rozdělování větších shluků na menší. Výsledkem je opět dendrogram, tentokrát od kořene (celá množina) k listům (jednotlivé objekty).

Dosud žádná z metod divizivních nemá zřetelnou výhodu proti aglomerativním. Rozdělení množiny na „podobnější si“ objekty vyžaduje náročnější výpočty. Někdy se uvádí, že dělení „shora“ najde rychleji několik základních shluků. Obvykle totiž nevyžadujeme celou hierarchii, ale vystačíme s několika shlukovacími úrovněmi. Ovšem dělení každého jednoho shluku se provádí vždy na 2 části, ty opět na 2 části atd. Vzniká proto obdobný efekt, jako při shlukování po dvojicích – při stejných shlukovacích hladinách se tyto informace ztrácí a může docházet ke zkreslení výsledku.

Uvedeme si algoritmus jedné z divizivních metod - metodu MacNaughtona-Smitha.

Algoritmus divizivní

1. Mějme shluk U.
2. Najdeme v něm bod K s největší průměrnou vzdáleností **vz** od ostatních bodů shluku. Tento bod tvoří základ nově vznikajícího shluku V, oddělujícího se od shluku U.
3. Hladina rozdělení v tomto okamžiku je rovna **vz/2**.
4. Dále ve shluku U najdeme bod, který má největší rozdíl vzdáleností G od zbývajících bodů shluku U a od všech bodů shluku V (obě vzdálenosti průměrné).

   Je-li $G > 0$, přemístíme tento bod do shluku V a proces opakujeme,

   $G \leq 0$, rozdělování shluku U ukončíme a za hladinu rozdělení dosadíme poloviční součet vzdáleností prvního neodděleného bodu od bodů shluku U a V.
5. Rozdělujeme-li shluk o dvou objektech, přiřadíme rozdělovanému shluku hladinu o hodnotě poloviční vzdálenosti jeho objektů.
6. Rozdělování shluků se opakuje, dokud existují shluky víceprvkové.

❑ Postup při shlukování reálných dat

Shrňme si na závěr doporučený optimální postup shlukování, máme-li analyzovat reálná data:

1. vybereme z dat reálné a ordinální atributy
2. zvolíme hledisko, ze kterého budeme hledat podobnost objektů
3. vybereme atributy charakterizující tuto podobnost
4. podle typu atributů zvolíme míru podobnosti či míru vzdálenosti objektů
   - koeficienty korelace, asociace
   - metriky
5. mají-li reálná data různé měrné jednotky či rozdílné domény, standardizujeme je
6. podle typu řešené úlohy zvolíme pojem „shluku“ jako optimalizovaný kulový nebo existující přirozený
7. podle typu řešené úlohy zvolíme typ rozkladu – prostý rozklad, hierarchie rozkladů, případně fuzzy-rozklad
8. hledáme-li kulové shluky a známe počet výsledných shluků,
   - zvolíme metodu k-středovou FORG se zadaným k
   - známe-li počáteční typické body, zadáme je, jinak pro ně volíme některou z metod TYPn
9. hledáme-li kulové shluky a neznáme počet výsledných shluků,

- zvolíme metodu k-středovou CLASS, k zadáme přibližně
- počáteční typické body zadáme některou z metod TYPn

10. hledáme-li přirozené shluky,
    - zvolíme aglomerativní metodu se strategií nejbližšího souseda; podporuje-li náš SW shlukování n-tic, případně zadání tolerance, zvolíme toleranci 10% ... dostaneme výsledek VYSL1
    - podle výsledného dendrogramu zvolíme jeden nebo několik optimálních rozkladů na shluky a reprezentanty těchto shluků jako jejich vnitřní body
    - zadáme známý počet shluků **k** i známé reprezentanty shluků jako typické body do metody k-středové FORG ... dostaneme výsledek VYSL2
    - na každé shlukovací úrovni porovnáme shluky obou výsledků
        - jsou-li oba výsledky stejné, množina našich objektů tvoří přirozené α-shluky dostatečně vzájemně vzdálené
        - liší-li se podstatně oba výsledky, množina našich objektů tvoří přirozené α-shluky, které nejsou od sebe příliš vzdálené nebo tvoří složité „nekulové“ tvary.
11. podle výsledků shlukovací analýzy a statistických charakteristik výsledných shluků interpretujeme charakteristické vlastnosti každého shluku; formulujeme je jako pravidla, podle kterých se objekt přiřadí do příslušného shluku
12. vyhodnotíme výsledek celé shlukovací analýzy

❑ Praktické příklady použití shlukovací analýzy

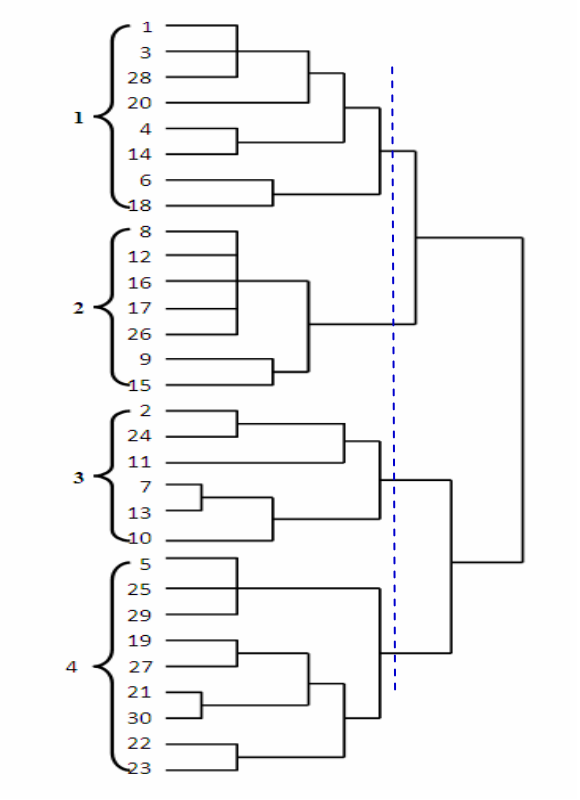
Příklad 5.27.

Během postgraduálního kurzu, kde byla mj. vykládána metoda GUHA, byla získána následující data. Objekty tvoří posluchači kurzu, data jsou získána z dotazníků, které posluchači vyplnili. Otázky dotazníku jsou několika druhů: jeden blok jsou obecné vlastnosti (pohlaví, věk, ...), druhý jejich psychologické vlastnosti (jste důvěřivý, ...), třetí pak jejich reakce na konkrétní hudební ukázky (autor dotazníku se mj. zabývá vztahem hudby a matematiky, zde výzkumem souvislostí mezi psychologickými vlastnostmi lidí a způsobem reakce na hudbu). Otázky jsou formulovány ano - ne, tedy data jsou binární. Posluchačů bylo 30, otázek 28. Hudební ukázky byly úryvky těchto skladeb: J.S.Bach: 25. variace z Goldbergovských variací pro cembalo, G.Mahler: začátek Adagia z 6.symfonie a A.Honegger: závěr 2. symfonie.

Otázky:	četnost A - N
1. Jste jedináček ?	6 - 24
2. Jste narozen před rokem 19xx ? (nad 35 let, pod 35 let)	26 - 4
3. Máte děti ?	20 - 10
4. Myslíte, že jste důvěřivý ?	16 - 14
5. Hrajete nebo hrál jste na nějaký hudební nástroj ?	17 - 13
6. Vadí vám neslušné vtipy ve společnosti ?	3 - 27
7. Věříte v nějaký vyšší řád nebo myslíte, že vše je dílem náhody ?	21 - 9
8. Máte rád rychlá rozhodnutí ?	19 - 11
9. Připadá vám ukázka Bacha radostná nebo smutná ?	8 - 22
10. Připadá vám ukázka Bacha vzrušená nebo klidná ?	21 - 9
11. Připadá vám ukázka Bacha blízká nebo cizí ?	26 - 9
12. Připadá vám ukázka Mahlera radostná nebo smutná ?	7 - 23
13. Připadá vám ukázka Mahlera vzrušená nebo klidná ?	7 - 23
14. Připadá vám ukázka Mahlera blízká nebo cizí ?	18 - 12

15. Připadá vám ukázka Honeggera radostná nebo smutná ?	26 - 4
16. Připadá vám ukázka Honeggera vzrušená nebo klidná ?	29 - 1
17. Připadá vám ukázka Honeggera blízká nebo cizí ?	10 - 20
18. Jste muž ?	27 - 3
19. Líbí se vám líčení u žen ?	17 - 13
20. Je víno vhodnější nápoj pro Čechy než pivo ?	10 - 20
21. Chodíte aspon jednou za dva měsíce na koncerty ?	6 - 24
22. Líbí se vám více červená než zelená ?	10 - 20
23. Líbí se vám více žlutá než modrá ?	9 - 21
24. Četl byste raději Tři muže ve člunu než Tři kamarády ?	16 - 13
25. Strávil byste volný večer raději sám doma nebo ve společnosti ?	16 - 13
26. Měl jste někdy při hudbě barevné představy ?	15 - 15
27. Měly souboje něco do sebe ?	25 - 5
28. Jste nápaditý ?	10 - 20

Shlukováním objektů – účastníků kurzu metodou hierarchickou a strategií nejbližšího souseda dostaneme dendrogram:



Na 3. shlukovací hladině od konce dostáváme 4 shluky přibližně stejně zastoupené. Po výpočtu charakteristik těchto shluků dostáváme:

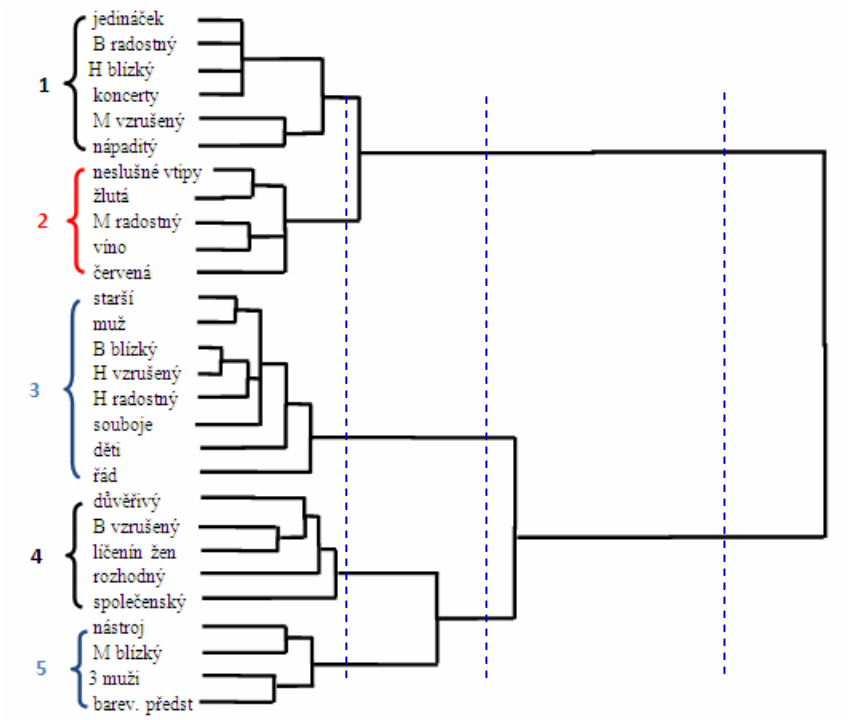
*Shluk 1**: **starší+muž**+děti+doma+řád*

*Shluk 2: nejed+nástro +**vtipy+rozhod+pivo**+nekoncert+**souboje+zelená**+modrá*

*Shluk 3: nejed+**starší**+děti+**nehraje+pivo+nebarev+nenápaditý**+nekoncert+ 3kamar+vtipy*

*Shluk 4: **muž**+starší+**líčení+nekoncert**+nenápaditý*

Stejná data můžeme nejprve transponovat a pak shlukováním dostaneme shluky podobných vlastností. Dendrogram metodou W-W:



Zřetelné jsou 2 shluky na poslední hladině, rozlišující především vlastnosti „veselejší, llehčí“ od „vážnějších, zodpovědnějších“. Skupina druhá se rozpadá vlastnosti „domácké“ a „společenské“, případně se v další vyznačené hladině rozpadají na 5 shluků popsaných na dendrogramu.

♦

Shrnutí pojmů 5.3.

Typy shlukovacích metod hierarchických. Metody aglomerativní a divizivní.

Základní algoritmus shlukování aglomerativního po dvojicích shluků.

Matice nepodobností - vzdáleností objektů.

Míra vzdálenosti shluků – shlukovací strategie. Strategie nejbližšího souseda, nejvzdálenějšího souseda, mediánová, centroidní, Ward-Wishartova.

Dendrogram, shlukovací strom.

Výsledné charakteristiky shluků a jejich interpretace.

Chyba algoritmu shlukování nejbližších dvojic.

Shlukování aglomerativní definitní, po n-ticích shluků.

Tolerance výpočtu při aglomerativním shlukování.

Shlukování divizivní, základní princip metod divizivních.

Interpretace výsledků shlukování.

Kompletní řešení shlukovací úlohy a jeho etapy.

Otázky 5.3.

1. Co je shlukování hierarchické?
2. Které typy hierarchických metod znáte a čím se liší?
3. Uveďte základní algoritmus aglomerativního shlukování po dvojicích.
4. Co je a k čemu je matice vzdáleností?
5. Co je vzdálenost shluků a které míry vzdálenosti znáte?
6. V čem je rozdíl mezi jednotlivými mírami vzdáleností shluků?
7. Proč může být výsledek aglomerativního shlukování po dvojicích nejbližších shluků chybný?
8. Uveďte základní algoritmus aglomerativního shlukování definitního.
9. Co je výsledkem shlukování hierarchického?
10. Co je dendrogram a k čemu slouží?
11. Které charakteristiky výsledných shluků je vhodné znát pro interpretaci výsledků?
12. Jak se pomocí výsledných charakteristik shluků provádí interpretace výsledků?

Úlohy k řešení 5.3.

1. Řešte data z kapitoly 5.1./1 BANKA jako hierarchickou úlohu, navrhněte, kterou metodou budete data shlukovat, případně navrhněte pro zvolenou metodu vhodné parametry.

2. Totéž proveďte pro data 5.1./2 GYMNÁZIUM.

3. Najděte alespoň 3 praktické úlohy, kdy bude vhodné použít pro dolování znalostí hierarchickou shlukovací metodu. Zvolte ke každé z nich míru vzdálenosti objektů, shlukovací strategii, toleranci „shody“ nejmenších vzdáleností a zdůvodněte tyto volby.

6. KLASIFIKACE

Čas ke studiu: 1 hodina

Cíl Po prostudování této kapitoly budete umět

- charakterizovat úlohy vhodné pro řešení rozhodovacím stromem
- popsat princip algoritmu konstrukce RS
- řešit praktické úlohy pomocí RS

Výklad

Co je rozhodovací strom

Jistou kombinací hledání shluků objektů a nalezení asociací mezi atributy je klasifikace objektů pomocí rozhodovacího stromu.

Na rozdíl od shlukování, které hledá shluky bez apriorní znalosti klasifikačních tříd, jsou zde tyto třídy dány. Hledají se podmínky ve formě hodnot vstupních atributů, za kterých padne objekt do jednotlivých klasifikačních tříd.

Na rozdíl od asociací nehledá, ale konstruuje optimální implikace mezi množinou vstupních atributů a výslednou klasifikací.

Rozhodovacím stromem se nazývá tato metoda proto, že výsledná pravidla se s výhodou zobrazují formou stromu. Uzly zobrazují atributy, podle nichž se právě dělí, hrany charakterizují jednotlivé hodnoty atributů.

Konstrukce rozhodovacích stromů je metodou získávání znalostí z dat, která v datech **hledá charakteristický popis zadaných tříd pomocí kombinací hodnot atributů.**

Patří k metodám induktivního učení s učitelem. To znamená, že z úplných dat (včetně hodnot klasifikační třídy) odvodíme pravidla. Ty pak můžeme použít i pro data, u nichž klasifikační třídu neznáme. Podle odvozených pravidel ji předpovíme.

Základní pojmy

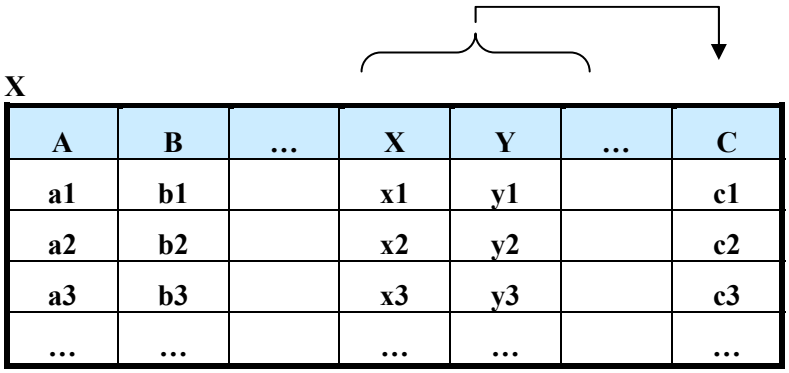
Je dána matice **X** o **m** objektech **O**={O_1 ,…, O_m} a **n** atributech **A**={A_1 ,…, A_n} s doménami **D**j={Dj0,…Djh}, množina tříd **C** = {C1, … Ck}; třídy jsou charakterizovány hodnotami jednoho nebo několika atributů **T** ⊂ **A**. Jedna nebo několik kombinací hodnot atributů z **T** definuje jednu třídu Ci.

Trénovací množinou T pro konstrukci konkrétního rozhodovacího stromu nazýváme takovou množinu objektů **O** = {O1, O2, …, Om} s atributy **A** = {A1, A2, …, An }, kde každý objekt je ohodnocen některou hodnotou klasifikačního atributu C.

Nazveme

atributy Ai ⊂ (**A-T**) **předpovídajícími** (~ vstupní, antecedenty)

atributy Aj ⊂ **T předpovídanými** (~ klasifikační, výstupní, sukcedenty)



A	B	…	X	Y	…	C
a1	b1		x1	y1		c1
a2	b2		x2	y2		c2
a3	b3		x3	y3		c3
…	…		…	…		…

Klasifikačním atributem nazýváme kategoriální atribut C, jehož hodnota určuje třídu objektu. Klasifikační atribut nemusí být primárně zadaným atributem, ale může být odvozen z kombinací hodnot předpovídaných atributů.

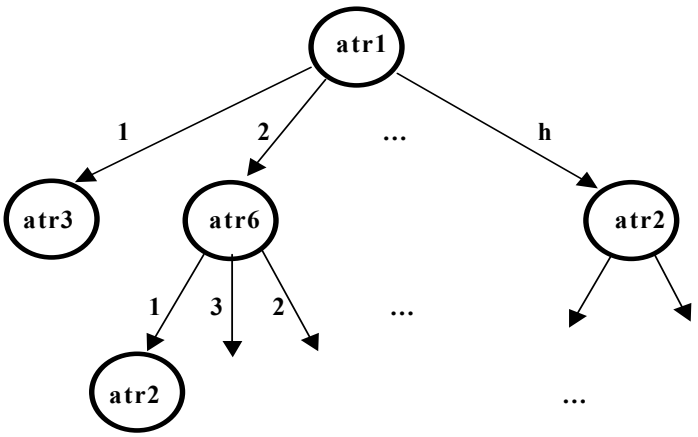
Rozhodovací strom je orientovaný acyklický souvislý graf – strom. Každý vnitřní uzel spolu s hranami z něj vycházejícími reprezentuje rozdělení trénovací množiny.

Každý list má přiřazenu hodnotu klasifikačního atributu, reprezentující nejpočetnější třídu objektů z podpůrné množiny listu. Nazýváme jej **klasifikační třídou listu.**

Podpůrná množina uzlu P_U je taková podmnožina trénovací množiny, do které patří objekty splňující podmínky rozdělení reprezentované vnitřními uzly a hranami na cestě od kořenu stromu k tomuto uzlu. Podpůrnou množinou kořene stromu je celá trénovací množina.

List je prostý, pokud všechny objekty jeho podpůrné množiny patří do stejné třídy.

Chyba listu je počet objektů podpůrné množiny listu, které nepatří do klasifikační třídy listu.



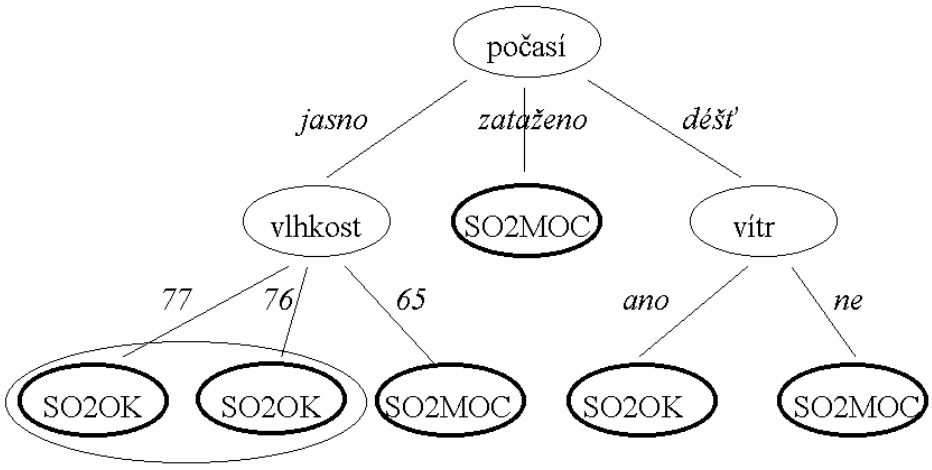
Obrázek 8.1. Rozhodovací strom

Příklad 8.1.

Použijeme často citovaný příklad Počasí [x]. *V datech z následující tabulky zaznamenané údaje o počasí, vlhkosti a větru chápeme jako předpovídající atributy, koncentraci oxidu siřičitého v ovzduší jako předpovídaný atribut. Jinak řečeno hledáme pravidla, pomocí nichž budeme na základě znalosti hodnot počasí, vlhkosti a větru předpovídat koncentraci SO2. Podle normy je maximální povolená hodnota SO2 = 150μg/m3, vyšší hodnota znamená překročení normy. Na základě toho určíme (odvodíme) z předpovídané hodnoty koncentrace SO2 dvě třídy SO2MOC a SO2OK.*

předpovídající			předpovídané	klasifik.. třída
počasí	vlhkost[%]	vítr	koncentrace SO2	
zataženo	78	ne	220	SO2MOC
déšť	80	ne	195	SO2MOC
jasno	76	ne	130	SO2OK
jasno	65	ano	200	SO2MOC
déšť	70	ano	115	SO2OK
déšť	80	ano	110	SO2OK
jasno	77	ano	120	SO2OK

Pomocí algoritmu popsaného níže dostaneme následující rozhodovací strom:



Obrázek 8.2. Strom předpovídání koncentrace SO2 z počasí

♦

Princip metody

Máme matici **X** o **m** objektech a **n** atributech a množinu tříd **C** = {C1, … Ck}.

Můžeme rozlišit dva případy:

- Všechny objekty z **X** patří do stejné třídy Ci (mají konstantní hodnotu posledního sloupce C). Pak rozhodovací strom pro **X** obsahuje jediný list, reprezentující třídu Ci.
- Objekty z **X** patří do více než jedné třídy C1, …, Ct. Pak je rozhodovací strom konstruován následujícím postupem. Vybere se jeden z antecedentů Ai, ten tvoří uzel stromu. Pro něj se provede rozklad τ uzlu tak, že každé hodnotě Dj atributu Ai je přiřazena jedna hrana označená hodnotou Dj, $j \in \{j1,\ldots, jk\}$. Rozklad rozdělí množinu **X** do podmnožin $\mathbf{X}_{j1}$, …, $\mathbf{X}_{jk}$ tak, že objekt Oi patří do podmnožiny **X**j právě tehdy, když Oi má v rozkladu τ výstup Dj.

Další růst stromu je rekurzivní aplikací bodů 1 a 2 na podmnožiny $\mathbf{X}_{j1}$, …, $\mathbf{X}_{jk}$. V bodě 2 se vybírá pro rozklad takový antecedent, který ještě nebyl použit v cestě od kořene stromu k aktuálnímu uzlu. Když není možno aplikovat bod 2, růst stromu se zastaví.

Dosud jsme blíže nespecifikovali, který z dosud nepoužitých atributů bude v konkrétním okamžiku vybrán pro další rozklad. Přitom tento výběr je rozhodující pro správnost klasifikace objektů.

Jeden z možných postupů je založen na informačním zisku rozhodovacího stromu při přechodu z jednoho stavu budování stromu do druhého, rozšířeného o další uzel. Informační zisk je měřen rozdílem entropie rozkládané podmnožiny objektů a váženého součtu entropií množin představujících výsledky rozkladu.

Konstrukce optimálního rozhodovací stromu

Nechť **X** je počáteční učící množina objektů a $X \subset \mathbf{X}$ je podmnožina, která má být v nějaké fázi budování rozhodovacího stromu rozložena rozkladem pomocí atributu $A \subset \mathbf{A}$. Nechť D1, …, Dk jsou výstupy tohoto rozkladu a platí pro ně:

$$Di \subseteq \mathbf{D}, \quad \cup Di = \mathbf{D}, \quad Di \cap Dj = \varnothing, \quad i \neq j,\ 1 \leq i, j \leq k$$

Označme frek (Cj, D) počet výskytů objektů třídy Cj v množině D,

|D| nebo |Di| počet objektů v množině D nebo Di.

Pro rozkládanou množinu **D**, předpovídající atribut **A** s doménou {D0, ... Dh} a klasifikační třídu **C** s doménou {C1, … Ck} je frekvenční tabulka

C \ A	**D_0**	**D_1**	**D_2**	**…**	**D_h**	
C_0	**$\lvert D_{00}\rvert$**	**$\lvert D_{01}\rvert$**	**$\lvert D_{02}\rvert$**		**$\lvert D_{0h}\rvert$**	**frek(C_0, D)**
C_1	**$\lvert D_{10}\rvert$**	**$\lvert D_{11}\rvert$**	**$\lvert D_{12}\rvert$**		**$\lvert D_{1h}\rvert$**	**frek(C_1, D)**
C_2	**$\lvert D_{20}\rvert$**	**$\lvert D_{21}\rvert$**	**$\lvert D_{22}\rvert$**		**$\lvert D_{2h}\rvert$**	**frek(C_2, D)**
…						
C_k	**$\lvert D_{k0}\rvert$**	**$\lvert D_{k1}\rvert$**	**$\lvert D_{k2}\rvert$**		**$\lvert D_{kh}\rvert$**	**frek(C_k, D)**
	$\lvert D_0\rvert$	**$\lvert D_1\rvert$**	**$\lvert D_2\rvert$**		**$\lvert D_h\rvert$**	**|D|**

Potom definujeme:

Entropie množiny D je dána vztahem (suma pro j = 1, …, k)

$$\text{info}(D) = -\sum (\text{frek}(Cj, D) / |D| * \log_2 (\text{frek}(Cj, D) / |D|))$$

Vážený součet entropií množin představujících výstupy rozkladu nad atributem A je

$$\text{info}_A(D) = \sum |Di| / |D| * \text{info}(Si)$$

Informační zisk rozkladu nad atributem A označujeme gain (A) a je měřen rozdílem entropií před rozkladem a po něm.

$$\text{gain}(A) = \text{info}(D) - \text{info}(Di)$$

Pro rozklad se vybere ten atribut, jehož použití vede k největšímu zisku informace.

❑ Převod rozhodovacího stromu na pravidla

Místo reprezentace výsledku pomocí rozhodovacího stromu je možno použít ekvivalentního zápisu pomocí množiny **produkčních pravidel**. Levé strany pravidel jsou tvořeny konjunkcemi hodnot atributů v cestě od kořene stromu k listu, který označuje jednu třídu z pravé strany pravidla. Majoritní třída (ta, která koresponduje s největším počtem listů) je prohlášena za **implicitní třídu** a tedy pravidla pro ni se neuvádí.

Výsledný rozhodovací strom lze mechanicky převést na pravidla typu

Jestliže α pak β se spolehlivostí ε

kde α je podmínka tvaru elementární konjunkce typu A1=a1 ∧ A2 = a2 ∧ …

β je určení příslušnosti ke klasifikační třídě C= ci,

ε je chyba listu (pravidla).

Příklad 8.2.

Tentýž výsledek z dat POČASÍ zapsaný formou produkčních pravidel:

implicitní třída = *SO2MOC*

if počasí = *jasno* ∧ vlhkost > 75 ⇒ **třída = *SO2OK***

if počasí = *déšť* ∧ vítr = *ano* ⇒ **třída = *SO2OK***

♦

❑ Algoritmy pro konstrukci rozhodovacích stromů

Algoritmů pro konstrukci RS existuje řada, většina z nich je variantami níže uvedeného základního algoritmu. Rodina těchto algoritmů bývá označována jako TDIDT (Top-Down Induction of Decision Trees), nejznámější z nich jsou ID3, C4.5 a C5 [3], [4].

Základní algoritmus je rekurzivní, konstruuje RS shora dolů.

1. Pokud všechny body trénovací (pod)množiny patří do stejné třídy, pokračuje se bodem 4.
2. Vybere se nejvhodnější atribut pro rozdělení trénovací (pod)množiny.
3. Objekty trénovací (pod)množiny se rozdělí na podmnožiny podle hodnot vybraného atributu.

4. Pro všechny vytvořené podmnožiny se opakuje postup od bodu 1.
5. Vytvoří se list a přiřadí se mu třída, do níž patří objekty podmnožiny.

Ačkoliv je algoritmus přirozeně rekurzivní, není obvykle možno použít rekurzivní implementaci. Pro rozsáhlejší data (s větším počtem atributů a jejich hodnot) se totiž velmi brzy vyčerpá paměť pro mezivýsledky rekurze. Proto je nutné algoritmy implementovat bez rekurze, což je výrazně náročnější.

Místo reprezentace výsledku pomocí rozhodovacího stromu je možno použít ekvivalentního zápisu pomocí množiny **produkčních pravidel**. Levé strany pravidel jsou tvořeny konjunkcemi hodnot atributů v cestě od kořene stromu k listu, který označuje jednu třídu z pravé strany pravidla. Majoritní třída (ta, která koresponduje s největším počtem listů) je prohlášena za **implicitní třídu** a tedy pravidla pro ni se neuvádí.

❑ Rozhodovací stromy nad velkými daty

Základní algoritmus je možno modifikovat například pro nekonzistentní nebo neúplná data nebo k potlačení upřednostňování atributů, jejichž test vede k triviálním výstupům.

Cílem konstrukce rozhodovacího stromu je získat optimální strom takový, že

1. svou strukturou nejlépe reprezentuje charakter zdrojových dat reprezentovaný objekty trénovací množiny,
2. obsahuje co nejmenší počet uzlů.

Důvody jsou zřejmé. Bez přesnosti by konstrukce stromu nedávala smysl a menší strom je přehlednější, srozumitelnější, lépe se interpretuje.

Základní algoritmus obě zásady dodržuje volbou optimálního atributu pro dělení v každém uzlu stromu. Přesto u rozsáhlých dat může být výsledný strom příliš rozsáhlý. Nejen se tak prodlužuje výpočet, ale výsledek je nepřehledný, špatně se interpretuje.

U rozsáhlých dat se proto používají techniky k omezení růstu stromu. Existují dvě základní techniky pro ovlivnění přesnosti a růstu stromu:

1. **omezení růstu stromu** během konstrukce; do algoritmu se vloží dodatečné podmínky, které rozhodují o dalším dělení uzlu
   a) definováním **minimální velikosti uzlu** pro dělení (malé uzly se již nedělí, i když nejsou homogenní vůči klasifikační třídě),
   b) definováním **minimálního zisku z dělení**,
2. pomocí **vícefázové konstrukce** výsledného stromu minimalizovat strom na optimální velikost pro interpretaci. V první fázi se zkonstruuje celý strom, ve druhé, případně dalších fázích se výsledný strom redukuje na optimální (z hlediska velikosti a přesnosti) velikost
   a) prořezáváním stromu, a to buď nahrazením podstromu listem, nebo zvednutím podstromu,
   b) optimalizací stromu převedeného na pravidla,
   c) konstrukcí a testováním; v případě velkého počtu zdrojových dat rozdělením dat na dvě části; první se použije ke konstrukci stromu, druhá pak k testování, zda strom rozhoduje správně; v případě chybného zařazení se strom opraví (například nahrazením listu podstromem).

❑ Využití rozhodovacích stromů

- Klasifikační strom slouží k odhalení závislostí klasifikačního atributu na vstupních atributech.
- Výsledná pravidla umožní automaticky zařazovat nové objekty do klasifikačních tříd.
- Výsledný strom či pravidla umožní rozlišit důležitost předpovídajících atributů, případně vybrat jen atributy významné pro zařazování do klasifikačních tříd; tak je možno snížit rozměr dat při zachování stejné informace.
- Výsledný strom umožní rozpoznat existenci shluků v datech (charakterizovaných výstupní třídou). Stromová struktura umožní shluky lépe interpretovat.

❑ Výhody a nevýhody konstrukce rozhodovacích stromů

Závěrem si přehledně zopakujeme výhody a nevýhody metody konstrukce rozhodovacích stromů.

Výhody

- Metoda je široce použitelná na mnoho typů dat.
- Základní algoritmus je jednoduchý, což umožňuje využití v mnoha oborech.
- Výsledky u menších stromů mají jednu z nejnázornějších reprezentací.
- Výsledek umožňuje klasifikovat i objekty v době konstrukce stromu neznámé.
- Rozhodovací strom lze snadno převést na pravidla.

Nevýhody

- Ne vždy jsou stromy schopny modelovat složitější reálné situace.
- U větších dat může velikost stromu komplikovat jejich interpretaci.
- Kvalita výsledných znalostí je silně závislá na tom, jak dobře a úplně pokrývá trénovací množina celou problematiku úlohy. Pokud trénovací množinu vytváří expert, dá se očekávat systematické pokrytí celé problematiky úlohy. Pokud máme k analýze observační data, nedá se obecně předpokládat nic, pokud data a jejich původ dobře neznáme. Obecně pak platí „čím více, tím lépe“, při větší trénovací množině je vyšší pravděpodobnost budoucí správné klasifikace nových objektů.
- Výsledek klasifikace je citlivý na neúplná data, kvalita se tím snižuje.
- Jedním z důvodů chybného výsledku může být také provedená kategorizace původních reálných dat – rozdělení do třídních intervalů napříč některými hodnotami klasifikační třídy.

Shrnutí pojmů 6.

Objekty a atributy. Atributy předpovídající a předpovídané. Klasifikační třída.

Trénovací množina.

Rozhodovací strom. Klasifikační třída listu. Podpůrná množina uzlu.

List prostý. Chyba listu.

Entropie, vážený součet entropií. Informační zisk.

Algoritmus konstrukce rozhodovacího stromu.

Výsledek výpočtu formou rozhodovacího stromu a formou rozhodovacích pravidel.

Implicitní třída.

Optimalizace rozhodovacího stromu pro rozsáhlá data.

Omezení růstu stromu.

Vícefázová konstrukce stromu. Prořezávání.

Konstrukce a testování stromu.

Otázky 6.

1. Co je rozhodovací strom a jakou úlohu nad daty řeší?
2. Jak se dělí atributy pro využití rozhodovacího stromu?
3. Co je entropie množiny objektů?
4. Co je vážený součet entropií množin představujících výstupy rozkladu nad atributem A?
5. Co je informační zisk?
6. Uveďte princip algoritmu pro konstrukci rozhodovacího stromu.
7. Jak se vybírají optimální atributy v průběhu konstrukce rozhodovacího stromu?
8. Jak se zobrazují výsledky výpočtu konstrukce rozhodovacího stromu?
9. Jak se výsledky zapisují formou produkčních pravidel?
10. Co je a jak se určí implicitní třída?
11. Kterými typy metod se řeší problémy s rozsáhlými daty a rozsáhlým výsledným stromem?
12. Jakými způsoby se omezuje růst stromu?
13. Co je vícefázová konstrukce stromu?
14. Jak je možno využít velkou trénovací množinu k urychlení konstrukce stromu?

Úlohy k řešení 6.

1. Jsou dána data PACIENTI s atributy

   INFARKT (ano/ne),
   ANG_PECTORIS (ano / ne),
   BERC_VRED (ano / ne),
   VAHA (kg),
   VYSKA (cm),
   KURAK (ano / ne),
   POHLAVI (muž / žena),
   VEK (roků),
   MESTO (ano / ne),
   DUCHODCE (ano / ne),
   STRES (ano / ne).

Data jsou pořízena za 10 let praxe praktického lékaře a obsahují 2300 objektů.

Navrhněte použití metody rozhodovacího stromu pro nalezení příčin infarktu, příp. angíny pectoris nebo bercového vředu.

7. NĚKTERÉ DALŠÍ ANALÝZY

Čas ke studiu: 3 hodiny

Cíl Po prostudování této kapitoly budete umět

- popsat metodu pro studium závislosti dvou reálných veličin a na praktických příkladech ji použít
- popsat metodu pro studium závislosti dvou kategoriálních veličin a na praktických příkladech ji použít
- popsat analýzu výjimek a na praktických příkladech ji použít

Výklad

7.1. Metoda CORREL

❑ Studie závislosti dvou reálných atributů

Pro určení lineární závislosti dvou reálných veličin používáme obvykle koeficient korelace. Víme, že pro nezávislé veličiny je výsledná hodnota koeficientu korelace blízká nule. Pro lineárně závislé veličiny se blíží některé z hodnot ±1. Graf závislosti obou veličin se pak blíží přímce.

Mějme opět datovou matici **X** s objekty {O1, O2, …} v řádcích a atributy {A1, A2, …} ve sloupcích. Otestujeme dvě veličiny – dva reálné atributy A1, A2 na celé množině objektů a dostaneme korelaci velmi blízkou nule, tedy považujeme obě veličiny za nezávislé.

X

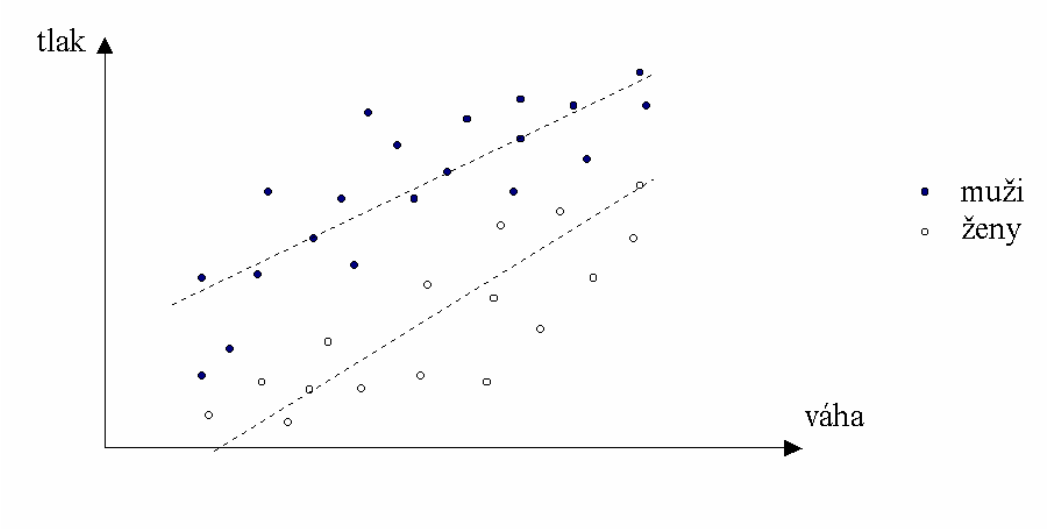
	A1	**A2**	**…**	**Aj**	**An**
O1	x11	x12			
	…	…			
Oi	xi1	xi2			
	…	…			
Om	xm1	xm2			

Může se však stát, že pro některé podmnožiny objektů jsou obě veličiny lineárně závislé a hodnota korelace se blíží ±1. Podmnožiny obvykle vybíráme pomocí hodnot některých ostatních atributů – jednoho (Aj = J) nebo konjunkce několika (Aj=J ∧ Ak= K ∧ …).

Pak můžeme konstatovat, že za podmínky P (odpovídající některému tvaru výše uvedené elementární konjunkce) jsou veličiny A1, A2 lineárně závislé.

Příklad 7.1.

Mezi atributy sledovaných lidí jsou také váha a tlak. *Jejich graf je na následujícím obrázku. Vidíme, že všechny body rozhodně nepřipomínají přímku. Ovšem když vybereme zvlášť muže a zvlášť ženy, obě podmnožiny se již přímce blíží výrazně více.*



Obrázek 9.1. Závislost váhy a tlaku dle pohlaví

♦

❑ Metoda CORREL

Jako u všech metod dolování znalostí i následující metoda netestuje jen předem formulované tvary hypotéz o možných závislostech, ale systematicky je generuje, testuje a vydává zprávu o daty podporovaných hypotézách.

Procedura CORREL generuje zajímavé pravdivé sentence tvaru

(A1 corr A2) / KONJ

kde A1, A2 jsou reálné veličiny, atributy,

KONJ je elementární konjunkce - podmínka

corr je některý z korelačních kvantifikátorů.

Při jednom výpočtu jsou veličiny A1, A2 pevné, podmínka se mění.

Vstupní parametry procedury obsahují :

- reálné veličiny A1 a A2, jejichž vztah chceme zkoumat,
- použitý korelační kvantifikátor a pro něj upřesňující parametry výpočtu:

  pro Kendallův kvantifikátor k-corr α kde α = hladina významnosti

  Spearmannův kvantifikátor s-corr$_\alpha$ α jako výše

  pořadově ekvivalenční e-corr s s = minimální frekvence pro podmínku
- povolený tvar podmínky obdobně jako u ASSOC:

  množina predikátů důležitých B (v elementární konjunkci alespoň jeden) a ostatních C,

pro množiny B a C ke každému predikátu povolený tvar pozitivní, negativní nebo obojí,

maximální povolenou délku podmínky.

❑ Korelační kvantifikátory pro reálná data

posuzují kladné závislosti dvou reálných veličin. Asociace používají korelační kvantifikátory založené na pojmu pořadí.

Předpokládejme, že data **X** obsahují objekty O1, O2, . . ., Om a pro ně dva reálné atributy t1 a t2. Dále, že pro žádné dva objekty O1, O2 neplatí O1(t1) = O2(t2) nebo O2(t2) = O1(t1). Pro libovolný objekt Oi definujme **pořadí R(i)** objektu vzhledem k t1 jako počet prvků množiny {Oj Î **X** | Oj(t1) £ Oi(t2)}. Podobně definujeme pořadí objektů Q(i) vzhledem k t2. Pak

1. **Spearmanův kvantifikátor** s-corr$_\alpha$ pro α Î (0, 0.5>

$$\textbf{s-corr}_\alpha\textbf{ (< t}_1\textbf{ , t}_2\textbf{ >) = 1} \quad \text{je-li} \quad \sum_{i=1}^{m} R(i)Q(i) \geq k_\alpha$$

kde k_α je vhodná konstanta. Pokud pozorované veličiny mají spojité rozložení, jde o test nulové hypotézy nezávislosti proti alternativní hypotéze o kladné závislosti. Hodnota 1 znamená kladnou závislost; jde o test na hladině α.

2. **Kendallův kvantifikátor** k-corr$_a$ pro a Î (0, 0.5>

k-corr$_a$ (< t$_1$, t$_2$ >) = 1 je-li **å sign(R(i)-R(j)(sign(Q(i)-Q(j)) ³ k$_a$**

pro i ¹j, kde k$_a$ je vhodná konstanta. Statistická interpretace jako výše.

3. **Pořadově ekvivalenční kvantifikátor** e-corr$_a$

e-corr$_a$ (< t$_1$, t$_2$ >) = 1 je-li **R(i) = Q(i)** pro i=1, . . . , m

Korelační koeficienty se používají v sentencích podmíněného tvaru

F1 q F2 / F

kde F1 a F2 jsou reálné veli4iny a F je binární formule.

Otázky 7.1.

1. Kterými metodami je možno analyzovat vzájemný vztah mezi dvěma atributy, pokud nabývají hodnot reálných?
2. Popište princip metody CORREL a rozdíl této metody proti statistickému výpočtu korelace dvou veličin.

Úlohy k řešení 7.1.

1. Formulujte algoritmus pro metodu CORREL.

2. Napište program, realizující tento algoritmus.
3. Najděte alespoň 3 praktické úlohy na využití metody CORREL.
4. Jsou dána data AUTA03 z Torontského statistického úřadu obsahují informace o nových typech aut uvedených na trh v roce 2003. Soubor obsahuje 93 nových typů aut od 25 výrobců uvedených na trh. Každé typ vozidlo je charakterizováno 19 atributy.

   1.**Výrobce** - kateg

0 = Acura	10 = Geo	20 = Subaru
1 = Audi	11 = Hyundai	21 = Toyota
2 = BMW	12 = Lexus	22 = Volkswagen
3 = Buick	13 = Lincoln	23 = Volvo
4 = Caddilac	14 = Mazda	24 = Saturn
5 = Chevrolet	15 = Mercedes-Benz	
6 = Chrysler	16 = Mitsubishi	
7 = Dodge	17 = Nissan	
8 = Eagle	18 = Oldsmobile	
9 = Ford	19 = Pontiac	

   2. **Třída** - dělení typů bylo převzato od výrobců aut

      0 = malé auto
      1 = sportovní auto
      2 = auto střední třídy
      3 = auto vyšší třídy
      4 = nákladní vozidlo

   3. **Minimální cena** = cena za základní model (bez doplňků)

   4. **Střední cena** = průměr minimální a maximální ceny vozu

   5. **Maximální cena** = cena za nadstandartní model (s maximálními doplňky)

   6. **Spotřeba ve městě** (v mílích na galon)

   7. **Spotřeba na dálnici** (v mílích na galon)

   8. **Vybavenost airbagy**: 0 = žádný, 1 = jeden (řidič), 2 = dva (řidič i spolujezdec)

   9. **Náhon**: počet kol, na které je náhon

   10. **Počet válců**

   11. **Objem motoru** (v litrech)

   12. **Maximální počet koňských sil**

   13. **Vybavenost automatickou převodovkou**: 0 = Ne, 1 = Ano

   14. **Kapacita nádrže** (v galonech)

   15. **Maximální počet pasažérů**

   16. **Délka** (v palcích)

   17. **Šířka** (v palcích)

   18. **Hmotnost** (v librách)

   19. **Vozidlo je výroby:** 0 = neamerické, 1 = americké

   Najděte možné analýzy těchto dat pomocí metody CORREL a formulujte, jaký typ výsledku může metoda nalézt.

7.2. Metoda COLLAPS

❑ Studie závislosti dvou kategoriálních atributů

Pro určení závislosti dvou kategoriálních atributů používáme obvykle frekvenční tabulku, případně z ní vypočtené koeficienty asociace. Ovšem jediný koeficient asociace opět není dostatečnou charakteristikou vztahu dvou kategoriálních veličin. Jednou z metod, která systematicky vyhledává možné vztahy mezi podmnožinami hodnot obou atributů, je metoda kolapsování. Tato metoda

- analyzuje podrobně vztah dvou kategoriálních atributů, vyhledává zdroje závislosti,
- provádí kolapsování (sečítání) hodnot několika řádků a sloupců původní kontigenční tabulky, aby našla nejsilnější vztahy mezi dvěma kategoriálními atributy
- generuje zajímavé pravdivé sentence tvaru

$$\mathbf{A1[Ki] \sim A_2[Kj]}$$

kde A1, A2 jsou zvolené atributy,

K1, K2 jsou koeficienty vzniklé z podmnožin hodnot kategorií atributů A1, A2 takové, že jejich numerické charakteristiky vyhovují určitým podmínkám, které říkají, že veličiny spolu v datech souvisejí.

Vstupem pro proceduru jsou složené dvouhodnotové veličiny A1[K1] a A2[K2] a z nich vypočtené frekvence a, b, c, d. Na ně se aplikuje některý asociační koeficient.

Název procedury plyne z toho, že frekvence a, b, c, d dostaneme sečtením - kolapsováním řádků a sloupců úplné frekvenční tabulky obou kategoriálních veličin.

Frekvenční (kontigenční) tabulka atributů A1[s hodnotami H1 ... Hh] a A2 [s hodnotami G1 ... Gg]:

A1\A2	G1	G2	…	Gg	
H1	a_{11}	a_{12}		a_{1g}	r1
H2	a_{21}	a_{22}		a_{2g}	r2
…					…
Hh	a_{h1}	a_{h2}		a_{hg}	rh
	k1	k2	…	kg	m

⇒

K1\K2	ano	ne	
ano	a	b	k
ne	c	d	l
	r	s	m

Metoda vyšetřuje **všechny dvojice neprázdných koeficientů K1 (podmnožin hodnot H1 ... Hh) a K2 (podmnožin hodnot G1 ... Gg)** a jako výsledek vydává všechny ty dvojice, které prošly testem pro zvolený kvantifikátor a zvolené parametry. Výsledky uspořádá podle hodnoty kvantifikátoru, od největšího.

Vstupní parametry upřesňující, co je pro řešitele zajímavé, a jaké požaduje výstupy:

- Atributy A1, A2, z nichž jsou odvozeny binární veličiny A1[K1] a A2[K2], pro něž se má procedura provést.
- Použitý asociační kvantifikátor a parametry upřesňující jeho použití:

  N = maximální povolená délka řešení, maximální počet hypotéz,

C = χ^2, kritická hladina
Q = omezení podílu charakteristiky nejlepšího a nejhoršího prvku řešení (o kolik může být poslední hypotéza horší, než první).

- Zda řešíme úlohu nehierarchickou nebo hierarchickou (viz níže).
- Informaci o požadovaných výsledcích. Základním výsledkem je blokový rozklad frekvenční tabulky s absolutními frekvencemi. Na požadavek řešitele je možno počítat další charakteristiky jako

řádkové relativní frekvence	$(a_{ij} / r_i) * 100$
sloupcové relativní frekvence	$(a_{ij} / k_j) * 100$
relativní frekvence	$(a_{ij} / m) * 100$
očekávané frekvence	$o_{ij} = r_i * k_j / m$
odchylky	$a_{ij} - o_{ij}$
standardizované odchylky	$(a_{ij} - o_{ij}) / \sqrt{o_{ij}}$

❑ Příklad použití metody COLLAPS

Příklad 9.3.

V datech o 592 sledovaných lidech jsou mj. atributy barva očí *a* barva vlasů. *Frekvenční tabulka těchto dvou atributů je*:

OČI \ VLASY	1 = černá	2 = brunet	3 = zrzavá	4 = blond	SUMA
1 = hnědé	68	119	26	7	220
2 = šedé	15	54	14	10	93
3 = zelené	5	29	14	16	64
4 = modré	20	80	17	94	215
SUMA	108	286	71	127	592

Kvantifikátor CHIQ (N=3, C=0, Q=0) se vypočte pro koeficienty.

OČI [1] ~ VLASY [1]OČI [1,2] ~ VLASY [1] OČI [1] ~ VLASY [1,2]
OČI [1] ~ VLASY [2]OČI [1,2] ~ VLASY [2] OČI [1] ~ VLASY [1,3]
...
OČI [2] ~ VLASY [1] OČI [1,3] ~ VLASY [1] OČI [1] ~ VLASY [2,4]
OČI [2] ~ VLASY [2] OČI [1,3] ~ VLASY [2] OČI [1] ~ VLASY [3,4]
...
OČI [3,4] ~ VLASY [3,4]

(ostatní kombinace jsou doplňkové a není je třeba počítat znovu).

Pak posloupnost výsledků uspořádaná podle výsledné hodnoty CHIQ je:

1. OČI [3, 4] ~ VLASY [4] tj. zelené nebo modré oči souvisí s blond vlasy
   OČI [1, 2] ~ VLASY [1,2,3] doplňkový tvar téže hypotézy, pro obě je
   v([3, 4], [4]) = **101.17**

2. OČI [4] ~ VLASY [4] v([4], [4]) = **99.35**

3. OČI [1] ~ VLASY [1,2,3] v([1], [1, 2, 3]) = **69.36**

Všechny tři hypotézy nejsou v rozporu, obecně s tmavšími vlasy souvisí tmavší oči.

Volbou první hypotézy dostáváme výsledný rozklad frekvenční tabulky:

OČI\VLASY	4=blond	1=čer	2=bru	3=zrz	
3 = zelené	16	5	29	14	
4 = modré	94	20	80	17	
1 = hnědé	7	68	119	26	
2 = šedé	10	15	54	14	
					592

⇒

O\V	4	1,2,3	
3,4	110	165	
1,2	17	296	
			592

♦

❑ Hierarchická varianta metody COLLAPS

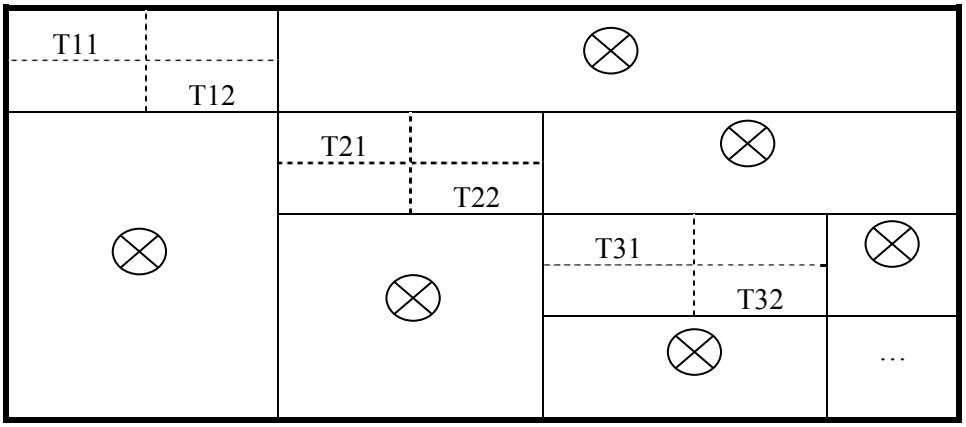
Procedura COLLAPS může řešit i hierarchickou úlohu.

Nehierarchickou úlohou rozumíme popsané vyhodnocení vztahu dvou atributů.

Hierarchická úloha znamená rekurzivní opakování této úlohy na podtabulkách četností, dokud podtabulky obsahují počet prvků > 1. V každém kroku se 2 nové podtabulky T1 a T2 tvoří z předcházející tak, že první obsahuje řádky a sloupce příslušné zpracovávaným veličinám K1 a K2, druhá řádky a sloupce ostatní.

Po prvním kroku (vyhledání K1, K2 s největší hodnotou kvantifikátoru) vytvoříme novou frekvenční tabulku tak, že vyškrtneme řádky s indexy nepatřícími do K1 a sloupce s indexy nepatřícími do K2. Takové podtabulky jsou dvě, základní T1 a doplňková T2.

Na tyto podtabulky aplikujeme znovu proceduru COLLAPS, dokud nejsou podtabulky jednořádkové nebo jednosloupcové a tedy je již nelze dělit.



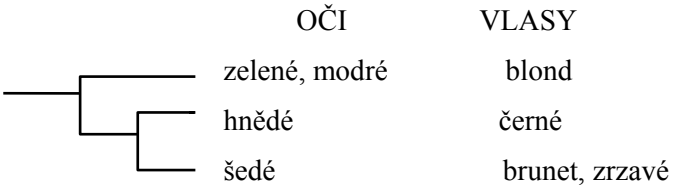
Výsledkem je pak binární strom nejlepších hypotéz, který vyjadřuje strukturu závislosti v tabulce.

Příklad 9.4.

V našem příkladě odpovídá plná čára rozkladu v 1. kroku. Krok 2 se neprovede, protože podtabulka T1 = [blond] je jednosloupcová. Krok 3 s rozkladem tabulky T2 = [černá, brunet, zrzavá] se provede (má 2 řádky a 3 sloupce), jeho výsledkem je rozklad označený přerušovanou čarou. Další kroky se neprovedou, protože další podtabulky jsou jednoprvkové.

OČI \ VLASY	4 = blond	1 = černá	2 = brunet	3 = zrzavá
3 = zelené	16	5	29	14
4 = modré	94	20	80	17
1 = hnědé	7	68	119	26
2 = šedé	10	15	54	14

Výsledek hierarchického rozkladu je vhodné vyznačit do stromové struktury, výsledkem je binární strom nejlepších hypotéz, vyjadřující strukturu závislostí v tabulce.



♦

❑ Uživatelem řízená hierarchie

Někdy může být vhodné ponechat na uživateli - analytikovi rozhodnutí, která hypotéza je nejzajímavější (nejsilnější) a které tabulky se mají dále rozkládat rekurzivní procedurou.

Příklad 9.5.

V našem případě může analytik například rozhodnout o výběru 2. hypotézy v 1. kroku. Obecná zkušenost totiž říká, že tmavší vlasy souvisí s tmavšíma očima a obě hypotézy mají jen velmi malý rozdíl hodnot CHIQ.

Pak by se rozkládala jiná podtabulka:

OČI \ VLASY	4 = blond	1 = černá	2 = brunet	3 = zrzavá
4 = modré	94	20	80	17
1 = hnědé	7	68	119	26
2 = šedé	10	15	54	14
3 = zelené	16	5	29	14

=

O\V	4	1,2,3
4	94	117
1,2,3	33	344

Pak z tabulky T2 vyjdou hypotézy:

1. hnědé oči černé vlasy 11.82
2. zelené oči zrzavé vlasy 7.64
3. zelené oči hnědé nebo zrzavé vlasy 6.73

První hypotéza má dostatečný odstup, zvolíme tedy ji a dostáváme rozklad:

OČI \ VLASY	4 = blond	1 = černá	2 = brunet	3 = zrzavá
4 = modré	94	20	80	17
1 = hnědé	7	68	119	26
2 = šedé	10	15	54	14
3 = zelené	16	5	29	14

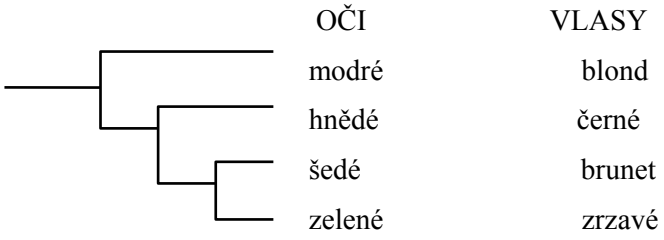
=

O\V	4	1	2,3
4	…	x	
1		68	145
2,3		20	111

Zbývá rozložit tabulku:

OČI \ VLASY	2 = brunet	3 = zrzavá
2 = šedé	54	14
3 = zelené	29	14

Celkově pak dostáváme výsledek, které odpovídá i naší intuitivní představě:



♦

Shrnutí pojmů 7.2.

Analýza vztahu dvou atributů kategoriálních.

Výpočet asociace pro generované podmnožiny hodnot atributů.

Hierarchie hodnot asociací dvojice atributů.

Otázky 7.2.

1. Kterými metodami je možno analyzovat vzájemný vztah mezi dvěma atributy, pokud nabývají hodnot kategoriálních?
2. Jaký je rozdíl mezi statistickým výpočtem CHIQ dvou veličin a metodou COLLAPS:
3. Jaký význam a využití má hierarchická metoda COLLAPS?

Úlohy k řešení 7.2.

1. Pro data AUTA03 (viz úloha 9.2./4) najděte zadání pro využití metody COLLAPS a formulujte význam případných výsledků.

7.3. Analýza výjimek

❑ Testování hypotéz prostřednictvím p-value

Úskalí klasických testů spočívá především v možnosti libovolného zvolení hladiny významnosti α. Nulovou hypotézu, kterou můžeme na jisté hladině významnosti zamítnout, lze při jinak zvoleném α přijmout. Libovolně zvolená hladina významnosti α tedy vede k libovolnému rozhodnutí. Tomuto problému se lze vyhnout, užijeme-li metodu testování hypotéz prostřednictvím p-value.

Konstrukce testu

1) stanovení nulové (H_0) a alternativní (H_A) hypotézy
2) volba výběrové statistiky a nulového rozdělení
3) určení hodnoty p-value
4) rozhodnutí o přijetí či zamítnutí nulové hypotézy

P-value

P-value je hodnota, která ukazuje jaká je shoda mezi daty a nulovou hypotézou. P-value lze definovat třemi různými způsoby, podle aktuálního tvaru H_A.

❑ *Jednostranné p-value*

H_A: parametr populace < nulová hypotéza

p-value: pravděpodobnost, že výběrová statistika bude alespoň tak malá, jako skutečně zjištěná hodnota, za předpokladu, že H_0 je pravdivá

$$p - value = \int_{-\infty}^{x} f(t)dt = F(x) \qquad (1)$$

kde *f(t)* je hustota pravděpodobnosti a *F(x)* distribuční funkce

HA: parametr populace > nulová hypotéza

p-value: pravděpodobnost, že výběrová statistika bude alespoň tak velká, jako skutečně zjištěná hodnota, za předpokladu, že H_0 je pravdivá

$$p - value = \int_{x}^{\infty} f(t)dt = 1 - F(x) \qquad (2)$$

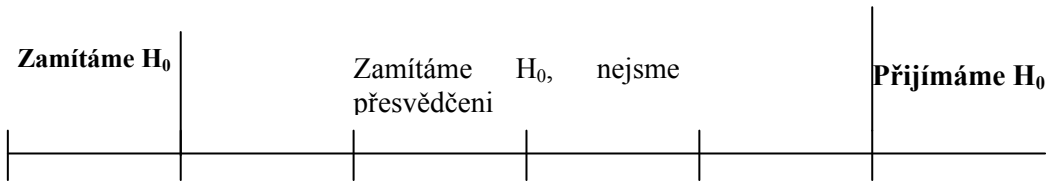
kde *f(t)* je hustota pravděpodobnosti a *F(x)* distribuční funkce

❑ *Oboustranné p-value*

p-value: pravděpodobnost, že výběrová statistika bude alespoň tak extrémní (tj. velká nebo malá) jako skutečně zjištěná hodnota, za předpokladu, že H_0 je pravdivá

$$p - value = 2 * \min\{F(x),\ 1 - F(x)\} \qquad (3)$$

Rozhodovací proces



Obr. 1: Rozhodovací proces p-value

V případě, že se p-value nachází v intervalu (0.01 až 0.05), doporučuje se opakování testu se zvětšeným rozsahem výběru (obecně – čím menší p-value, tím menší je platnost H_0).

❑ Konstrukce testu použitá pro analýzu výjimek

1) Stanovení nulové hypotézy H_0, a alternativní hypotézy H_A

   a) pro jednovýběrový test

   $H_0 : p_T = p_Z$

   $H_A : p_T \neq p_Z$

   kde p_T je relativní četnost v testovaném (výběrovém) souboru, a p_Z relativní četnost v základním souboru

   b) pro dvouvýběrový test

   $H_0 : p_1 = p_2$

   $H_A : p_1 \neq p_2$

   kde p_1 relativní četnost v prvním testovaném (výběrovém) souboru, a p_2 relativní četnost ve druhém testovaném souboru

2) Určení testovacího kriteria (testovací statistiky)

   Jako testovací statistiku zvolíme

   a) pro jednovýběrový test

$$T(\overline{X}) = \frac{p_T - p_Z}{\sqrt{p_Z(1-p_Z)}}\sqrt{n} \qquad (4)$$

   kde n značí rozsah výběru (počet prvků testovaného souboru).

   b) pro dvouvýběrový test

$$T_2(\overline{X}) = \frac{p_1 - p_2}{\sqrt{p(1-p)(\frac{1}{n_1}+\frac{1}{n_2})}} \qquad p_1 = \frac{x_1}{n_1};\ p_2 = \frac{x_2}{n_2};\ p = \frac{x_1 + x_2}{n_1 + n_2} \qquad (5)$$

   kde x_1, resp. $x2$, značí počet jednotek výběru mající specifikovanou vlastnost (nazývaný často jako „počet úspěchů“ a pak $n_{1(2)} - x_{1(2)}$ jako „počet neúspěchů“) v prvním, resp. druhém testovaném souboru. Dále n_1, resp. n_2, značí rozsah výběru u prvního, resp. druhého testovaného souboru.

Obě testovací statistiky mají dle [4] normální normované rozdělení *N(0, 1)*. Jedná se o jeden z jednovýběrových, resp. dvouvýběrových testů, používaných pro testování hypotéz o četnostech kategorií [5]. Testujeme, zda se statisticky liší střední hodnoty některého z prvků v testovaném souboru a základním souboru při jednovýběrovém testu, resp. mezi dvěma testovanými soubory (výběry) při dvouvýběrovém testu.

3) Nulové rozdělení F_0

$$F_0(x) = P(T(\bar{x}) < x \mid H_0) = \Theta(x) \qquad (6)$$

4) Určení hodnoty p-value

$$p - value = 2 * \min\{F_0(x),\ 1 - F_0(x)\}$$

5) Rozhodovací proces

Při zvolené hladině významnosti α=0.05 (jinak řečeno pravděpodobnost chyby 1.druhu je 5 %, což znamená že v 95 případech ze 100 se zamítla nulová hypotéza oprávněně. U zbylých nejvýše 5 % případů byla zamítnuta neoprávněně a ve skutečnosti platí. Se spolehlivostí 95 % nebo větší bylo přijato správné rozhodnutí) rozhodneme podle hodnoty p-value o přijetí či zamítnutí nulové hypotézy. Pokud tedy p-value > 0.05, pak přijímáme nulovou hypotézu. Pokud hodnota p-value leží v intervalu 0.01 – 0.05, jedná se o nepřesvědčivou oblast. Při hodnotě p-value < 0.01 (resp. p-value < 0.001), zamítáme nulovou hypotézu (resp. zamítáme H_0 ještě „silněji“). Výsledek výpočtu může tedy nabývat jedné ze šesti hodnot:

1. p-value < 0.001 (zamítáme H_0) a pravděpodobnost prvku základního souboru je větší než pravděpodobnost prvku testovaného souboru ($p_Z > p_T$) pro jednovýběrový test, resp. pravděpodobnost prvku druhého testovaného souboru je větší než pravděpodobnost prvku prvního testovaného souboru ($p_2 > p_1$) pro dvouvýběrový test; označíme ji znakem (--)
2. p-value < 0.01 (zamítáme H_0) a pravděpodobnost prvku základního souboru je větší než pravděpodobnost prvku testovaného souboru ($p_Z > p_T$) pro jednovýběrový test, resp. pravděpodobnost prvku druhého testovaného souboru je větší než pravděpodobnost prvku prvního testovaného souboru ($p_2 > p_1$) pro dvouvýběrový test; označíme ji znakem (-)
3. p-value < 0.01 (zamítáme H_0) a pravděpodobnost prvku základního souboru je menší než pravděpodobnost prvku testovaného souboru ($p_Z < p_T$) pro jednovýběrový test, resp. pravděpodobnost prvku druhého testovaného souboru je menší než pravděpodobnost prvku prvního testovaného souboru ($p_2 < p_1$) pro dvouvýběrový test; označíme ji znakem (+)
4. p-value < 0.001 (zamítáme H_0) a pravděpodobnost prvku základního souboru je menší než pravděpodobnost prvku testovaného souboru ($p_Z < p_T$) pro jednovýběrový test, resp. pravděpodobnost prvku druhého testovaného souboru je menší než pravděpodobnost prvku prvního testovaného souboru ($p_2 < p_1$) pro dvouvýběrový test; označíme ji znakem (++)

   První čtyři hodnoty tedy představují oblast zamítnutí nulové hypotézy, přičemž první dvě (1, 2) hodnoty odpovídají statisticky významně méně častému výskytu „úspěchu“ (jednotky výběru, mající specifickou vlastnost) v datech, a hodnoty 4 a 5 odpovídají statisticky významně častému výskytu „úspěchu“ v datech.
5. p-value leží mezi 0.01 a 0.05. Jedná se o nepřesvědčivou oblast viz. výše; označíme ji znakem (~)
6. p-value > 0.05 (přijímáme H_0), označíme ji znakem (.)

❑ Využití statistiky při dolování znalostí

Analýza výjimek je postupem, kdy se na základě porovnávání množiny základního a testovaného souboru hledají statisticky významné výskyty (ne)úspěchů, výjimek.

Základním souborem pro dolování dat, tedy i pro analýzu výjimek, rozumíme velké množství strukturovaných údajů o nějakém výseku reality. Tato množina dat bývá konečná a můžeme o ní tvrdit, že je pořízena statisticky náhodným výběrem. Nevíme tedy předem nic o pravděpodobnostech, rozděleních atd., a ve výpočtech budeme především používat četnosti sledovaných znaků.

Příkladem analýzy výjimek je hledání důvodů přerušení léčebného cyklu v databázi klinického registru léčebných cyklů centra asistované reprodukce (blíže viz. kapitola 7).

Vlastností této metody je to, že s využitím matematické statistiky, konkrétně testování hypotéz prostřednictvím p-value, automaticky vyhledává výjimky, na základě kterých pak vyslovíme určité hypotézy. Můžeme pak říct, že takovéto hypotézy buď přijímáme nebo zamítáme. Hypotézy vytvářené metodou analýzy výjimek jsou ve tvaru:

„statisticky významně častější výskyt výjimky v datech byl zaznamenán u“

- *1. nalezená výjimka*
- *2. nalezená výjimka*
- *...*

Použití analýzy výjimek je vhodné pro řešení výzkumných problémů na základě rozsáhlých dat. Analyzujeme data, protože se s jejich pomocí chceme něco zajímavého dozvědět. Zkoumáme problémy, které jsou definovány specificky, kdy na začátku máme položenu určitou otázku: *„co je příčinou výjimky v datech“*, a data jsou cíleně rozdělena tak, aby tato otázka byla zodpovězena.

Příklad 9.6.

Dlouhodobě sbíraná data o asistované reprodukci (umělém oplodnění) obsahují 8500 záznamů o provedených léčebných cyklech. Z nich jen 170 případů má v binárním atributu „hyperstimulační syndrom“ hodnotu 1. Jde tedy jen o 2% případů a v rámci předzpracování by se tento atribut za normálních okolností vyřadil ze zpracování. Pro lékaře by však bylo velmi důležité poznat příčiny vzniku tohoto syndromu. V rámci generování asociací nevyšly žádné výsledky.

Použijeme analýzy výjimek. První výběrový soubor tvoří záznamy bez syndromu,druhý soubor oněch 170 záznamů se syndromem. Analyzují se všechny atributy (kategoriální nebo kategorizované), které teoreticky mohou mít vliv na vznik syndromu – antecedenty.

Pro oba soubory se spočítají relativní četnosti všech hodnot antecedentových atributů, pro každou si odpovídající dvojici se spočítá hodnota p-value. Hodnoty vykazující statistickou odchylku se generují jako výsledky spolu s vyznačení nad- nebo podprůměrnosti hodnoty v testovaném souboru.

Výsledky si shrneme v tabulce, z níž je patrno, které atributy a s kterými hodnotami vykazují statistickou odchylku a tedy mohou být potencionální příčinou syndromu (nevysvětlujeme zde některé lékařské termíny, jako typ stimulace apod., jsou určeny k posouzení odborníkům).

Výskyt ovariálního hyperstimulačního syndromu (OHSS)

Atribut	OHSS (++)	OHSS (--)
Věk pacientky	mladší 25 let	Starší 30 let
OHSS dříve	1 a vícekrát	Dosud ne
Délka cyklu	> 30 dnů	< 21 dnů
Délka menstruace	7 – 9	
Operace dříve	0	1
Idiopat.faktor	Ano	Ne
Imunolog.faktor	Ano	Ne
Stimulace-typ	FSH/FSH+GnRH, HMG+GnRH	
Hodnota E2	> 10	< 10
Den cyklu AS	> 15	10 – 14
Získaných oocytů	> 8	0,1

♦

Shrnutí pojmů 7.3.

Analýza výjimek.

T test, hodnota p-value.

Generování hypotéz o hodnotách kategoriálních atributů, vykazujících statistické odchylky od základního souboru.

Otázky 7.3.

1. Který typ „zajímavosti“ v datech vyhledává analýza výjimek?
2. Popište princip analýzy výjimek a uveďte, pro která data je vhodná.

Úlohy k řešení 7.3.

1. Najděte alespoň 3 praktické úlohy na využití analýzy výjimek, definujte k nim výjimku a atributy zkoumané.

8. EXPERTNÍ SYSTÉM NAD DATOVOU MATICÍ

Čas ke studiu: 3 hodiny

Cíl Po prostudování této kapitoly budete umět

- popsat podstatu expertního systému, založeného na datové matici objektů a jejich konstruktů
- analyzovat, ladit a řešit praktické úlohy pomocí tohoto typu expertního systému

Výklad

8.1. Psychologický prostor člověka

Psychologický prostor, objekty a konstrukty

Psychologové zkoumají, jak lidé rozpoznávají a třídí zkušenosti a klasifikují své okolí a jak na základě toho předvídají budoucí jevy a řídí své jednání. Americký psycholog G. A. Kelly popsal postupy, kdy člověk vytváří svou strukturu pojmů (tzv. **psychologický prostor)** a model svých znalostí jako tzv. systém osobních konstruktů.

Osobní konstrukty jsou šablony, které si člověk vytváří pro klasifikaci objektů reality a potom se do nich snaží zasadit skutečnosti, ze kterých se skládá svět.

Používá k tomu dva pojmy: **objekty** jako oblasti zájmu (osoby, zvířata, věci, jevy, akce) a **konstrukty** jako vlastnosti objektů (atributy, znaky, ...):

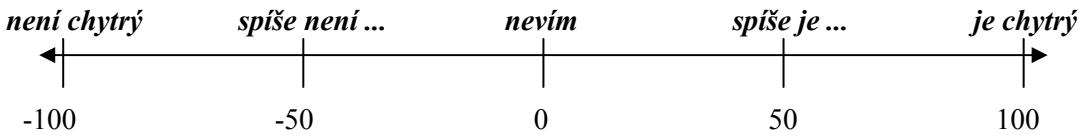
Konstrukt je tvořen dvěma protikladnými vlastnostmi, póly, které tvoří opačné mezní hodnoty atributu. Tím konstrukt svým způsobem normalizuje.

Psychologická vzdálenost mezi póly je rozdělena na stupně vyjadřující míru příslušnosti objektu k jednomu či druhému pólu. Střed škály znamená nevím, netýká se, proto je vhodné používat vícehodnotové stupnice s lichým počtem stupňů.

Rozdíl mezi atributem a konstruktem je právě v normalizaci konstruktu do škály <-100, 100>. Konstrukt je tedy zvláštním případem obecného atributu.

Příklad 10.1.

Objekty jsou studenti, *konstrukty např.* chytrý, pracovitý, *..., nejsou zde podstatné atributy jako* rodné číslo, ...



♦

Pro jemnější rozlišení informací pro uživatele se zavádí další parametr, **důležitost konstruktu**, kterou uživatel modifikuje vliv jednotlivých konstruktů na rozhodování při provádění konzultace:

Příklad 10.2.

1. *vybíráme-li studenty pro brigádu na tenisových kurtech, pak konstrukt* je chytrý *má důležitost 0*
2. *vybíráme-li studenty pro SVOČ, pak. konstrukt* je chytrý *má důležitost 100.*

♦

❑ Expertní systémy založené na pravidlech a založené na datech

Expertem v nějaké oblasti reality nazýváme člověka, který tyto oblast zná podstatně lépe, než je běžný průměr. Z člověka se stane expert tak, že teoretickým studiem nebo praktickým zkoumáním světa nebo získáváním znalostí dalšími způsoby pozná část reality dostatečně důkladně.

Aby se expertovy znalosti využily i ku prospěchu ostatních lidí, uděluje expert svou práci nebo rady ostatním (lékař rozpozná nemoc a léčí, učitel rozpozná neznalost a naučí apod.). Jedna z oblastí umělé inteligence – konstrukce expertních systémů – se snaží znalost experta vložit vhodným způsobem do počítače a řešení problémů či rady uživatelům pak nacházet automatizovaně.

Zjednodušeně řečeno, nejčastější způsob uložení znalostí experta do báze znalostí je formou pravidel tvaru „jestliže platí … pak …“

kde pravidla jsou právě získána od experta. Podmínky na sebe mohou navazovat, v programu je zabudován inferenční mechanizmus. Vytváří tak rozvětvené stromy, které modelují expertův způsob myšlení.

Na rozdíl od těchto klasických expertních systémů, které se snaží modelovat expertovy myšlenkové pochody, náš popsaný přístup modeluje expertovu paměť (modelují zkoumanou tématickou oblast jako psychologický prostor objektů a konstruktů).

8.2. Metodologie konstrukce expertního systému

❑ Realizace psychologického prostoru repertoárovou tabulkou

Uvedli jsme si, že expertní znalosti do databáze se přenáší pomocí **interview s expertem.** Expert tím pomáhá budovat bázi znalostí. Experta chápeme jako učitele, který předává systému své znalosti v podobě modelu svého psychologického prostoru objektů a konstruktů a prostřednictvím repertoárové tabulky průběžně sleduje, jak jeho „žák“ (báze znalostí) „chápe“ poskytnuté expertní zkušenosti. Učení je ukončeno, když báze znalostí poskytuje řešení podle jeho představ.

Zobrazením psychologického prostoru je **repertoárová tabulka**, kde v řádcích jsou objekty {O1,O2,…} a ve sloupcích konstrukty {K1, K2, …}

Objekt \ Konstrukt	*K1*	*K2*	*. . .*	*Kn*
O 1	*50*	*-50*		*-100*
O 2	*75*	*0*		*0*
. . .				
❑ **O m**	*100*	*-100*		*-75*

Hodnoty v tabulce znamenají, do jaké míry přísluší odpovídajícímu objektu jeden nebo druhý pól příslušného konstruktu.

Cílem je vytvořit z tabulky **bázi znalostí**, která bude řešit analytické úlohy podobně jako expert. To znamená vyplnit tabulku tak, aby dobře pokrývala celou modelovanou oblast zájmu – obsahovala dostatek příslušných objektů, charakterizovaných vhodnými konstrukty.

Příklad 10.3.

Uveďme jako příklad „expertní znalosti" pracovníka cestovní kanceláře, který doporučuje svým klientům, kam se mají vypravit na dovolenou v závislosti na jejich požadavcích, jako jsou: roční období, finanční náročnost, sportovní a společenské požadavky apod. Souhrn takových požadavků – konstruktů je vstupem pro zadání, výstupem pak jsou konkrétní doporučená místa – objekty. Přitom každé z míst není charakterizováno konstruktem jen binárně ano/ne, ale mírou příslušnosti (Alpy jsou spíše drahé, Beskydy spíše levné).

♦

Bude to báze s jednou vrstvou pravidel, pravidla směřují od dotazovacích uzlů (předpokladů), kterými jsou osobní konstrukty, k cílovým uzlům (závěrům), kterými jsou objekty. V našem případě se bude báze skládat z pravidel typu:

Jestliže se vyskytuje vlastnost K, pak jde o objekt O s vahou V

❑ Automatizace budování expertního systému

Všechny fáze práce s psychologickým prostorem člověka je možné automatizovat, realizovat pomocí vhodného programového systému. Jeho úkolem je

1. zprostředkovat dialog mezi expertem s jeho osobním psychologickým prostorem a repertoárovou tabulkou
2. vytvořit z repertoárové tabulky bázi znalostí ve formě soustavy pravidel,
3. pomocí báze znalostí podávat konzultace dalším uživatelům.

❑ Postup při budování expertního systému

1. Zadavatel vybere oblast z reality, které se bude expertní systém týkat, vybere experta na zvolenou oblast.
2. Expert zadá seznam všech možných **řešení** problému, kterého se bude vytvářená báze znalostí týkat. Prvky seznamu reprezentují závěry, které by měl poskytovat budoucí expertní systém (cíle konzultace, objekty).
3. Expert předem provede analýzu problémové oblasti, **vybere objekty**, které budou zkoumanou oblast reality dobře popisovat. Objekty mají být téhož typu, stejné úrovně složitosti a pokrývat téma co nejúplněji.
4. Expert vybere možné **konstrukty** objektů, jejichž póly by objekty dobře rozlišovaly.
5. Program vhodnými dotazy získává od experta informace o jeho psychologickém prostoru a sestavuje podle něho repertoárovou tabulku. Průběžně ji testuje a o výsledcích podává expertovi informace. Program analyzuje tabulku ze tří hledisek:
   - testuje postačitelnost informací pro popis objektů,
   - testuje vzájemné vazby mezi konstrukty,
   - vyhledává ekvivalentní nebo nedůležité informace.

6. Expert na každé upozornění dále upřesňuje poskytnuté informace a tak dále zjemňuje a rozšiřuje repertoárovou tabulku.
7. Program vytváří z každého políčka tabulky dvě pravidla, protože spojuje pravidlem každý pól každého konstruktu s každým objektem (mimo políčka „nevím“).

❑ Postup při využívání expertního systému

1. Uživatel vybere nebo zadá typ konzultace, vybere existující nebo zadá nový dotaz na expertní systém.
2. Při konzultaci uživatele s programem prochází systém všechny konstrukty a ptá se na jejich důležitost pro uživatele. Z hodnot políček repertoárové tabulky vypočítává váhu jednotlivých jednoduchých pravidel.
3. Příspěvky všech pravidel pro každý objekt sloučí a dospěje k závěru o vhodnosti jednotlivých objektů pro danou soustavu vlastností.
4. Výsledkem je seznam všech objektů báze s doporučením vhodnosti pro zadaný dotaz - pro každý objekt určí hodnocení z intervalu <-100,100>, kde -100 znamená naprosto nevyhovuje, 100 naprosto vyhovuje.

❑ Tvorba báze znalostí tedy probíhá v následujících krocích

1. Expert zadá seznam všech možných řešení problému, kterého se bude vytvářená báze znalostí týkat, prvky seznamu reprezentují závěry, které by měl poskytovat budoucí expertní systém (cíle konzultace, objekty);
2. pro dané cíle systém vytváří jejich trojice a žádá experta, aby pro každou trojici určil znak odlišující dva prvky od třetího; znak je určen pomocí dvojice opačných hodnot, pólů; vzniká tabulka ohodnocení, ve které je každý cíl popsán pomocí znaků, konstruktů; hodnota znaku se zadává s možností neurčitosti (ano, spíše ano, ..., nevím, ..., spíše ne, ne)
3. pro vytvořenou tabulku ohodnocení systém konstruuje znalosti ve formě implikací mezi jednotlivými póly různých znaků;
4. generují se pravidla, nejprve cílová (na levé straně se vyskytuje některý ze závěrů), potom mezilehlá (pro každou implikaci nalezenou v bodě 3); z každého pole tabulky ohodnocení se vygeneruje jedno pravidlo spolu s faktorem jistoty;
5. po vytvoření všech pravidel začíná testování báze znalostí; pokud expert nesouhlasí s výsledky konzultace, provádí se další zjemňování báze znalostí - přidáním nových znaků, nových objektů, změnami v tabulce ohodnocení ap.).

❑ Využití programového systému

- podstatně zkracuje dobu vytváření báze znalostí,
- po vytvoření slouží jako expertní systém vhodný pro výběr optimální varianty z množiny zaznamenaných variant nebo
- k automatizaci řízení systémů, u nichž je předem známo, co způsobí určitý řídicí zásah na výstupu systému (dopředné řízení).

8.3. Automatizované získávání expertíz SAZZE

Popsané principy realizace expertního systému nad datovou maticí byly využity pro implementaci programového systému, nazvaného SAZZE (**S**ystém **A**utomatizovaného **Z**ískávání **Z**nalostí od Experta).

❑ Fáze definování a naplňování repertoárové tabulky

Systém sérií otázek postupně od experta získává a ukládá tyto informace.

Název: NÁZEV BÁZE

Cíl: Určení a využití báze znalostí

Objekty: Konkrétní typy objektů

Konstrukty: Charakteristika konstruktů

Následuje dialog, ve kterém se systém postupně ptá na

1. Zadané objekty (vyplní levý sloupec repertoárové tabulky).

2. Zadané konstrukty (vyplní hlavičku – první řádek repertoárové tabulky).

3. Zadání hodnot konstruktů pro všechny objekty (vyplní sloupce a řádky repertoárové tabulky).

Výsledkem tohoto dialogu je repertoárová tabulka s hodnotami:

Potom z každého pólu každého konstruktu zformuluje systém cílová pravidla.

V rámci budování báze znalostí systém dále umožňuje provádět analýzu implikací a test shody objektů nebo konstruktů.

4. Analýza implikací

znamená nalezení všech dvojic logicky totožných implikací, které podle informací v bázi mohou platit. Analýzu provede systém a výsledky zobrazí. Expert pak potvrdí všechny skutečně platné. Ty jsou uloženy a budou využívány při konzultacích.

Expertem potvrzené implikace tvoří mezilehlá pravidla expertního systému.

5. Shoda objektù

znamená nalezení objektù, které mají ve všech konstruktech stejné hodnocení, porovnání se provádí procentuelně. Ve výsledku se uvádějí shody dvojic objektů nad 80%. Na dotaz systému může expert zadat nový konstrukt, který oba objekty rozliší.

Procentuelní shoda dvou objektů Oi a Oj (z celkového počtu n objektů popsaných m konstrukty) se vypočítá podle vztahu

$$S_{ij} = \sum_{a=1}^{n} \left(1 - \frac{|V_{ia} - V_{ja}|}{2 \times V_{\max}} \right) \times \frac{100}{n}.$$

kde x_{ik} je hodnota konstruktu k u objektu Oi.

6. Shoda konstruktů

Obdobně se mohou objevit podobné póly některých konstruktů, **shoda konstruktů**. Řešení je obdobné, jako u shody objektů, za hranici se bere 70%.

Sloučení konstruktů: pokud je shoda konstruktů dána nevhodně zvolenou skladbou konstruktů, je vhodné tyto konstrukty sloučit do jednoho. Jeho jméno mùže být shodné s jedním z původních nebo je možno zadat nové.

7. Konzultace s expertem

Po vytvoření báze znalostí expert konzultacemi ověřuje, jak se shodují doporučení expertního systému s jeho vlastními a v případě neshody modifikuje bázi tak dlouho, dokud systém nedává odpovědi stejné, jako on.

Každá konzultace má svůj název a je uloženo její zadání - požadavky, důležitost a hodnota konstruktů. Je možno vybrat z již existujících nebo definovat novou.

Uživatel vlastně zadává požadované hodnoty vlastností hledaného objektu, program vyhodnotí podobnost každého objektu tabulky se zadáním.

8. Zlepšování chování báze

Celkově má expert ladit bázi znalostí těmito nástroji:

- Přidáváním nových objektů
    - podle nápovědy, dotazem na nový objekt s hodnotami konstruktů, jejichž kombinace v bázi nejsou,
    - bez nápovědy
- Přidáváním nových konstruktů
    - bez nápovědy
    - podle nápovědy využitím mezilehlých pravidel (výsledku analýzy implikací), pro neplatná doplnit konstrukty s hodnotou různou u doporučených objektů a u nedoporučených objektů,
    - podle nápovědy s využitím výsledku konzultace, dotazem na nový konstrukt s hodnotami různými pro expertem doporučené a nedoporučené objekty, přičemž výsledek konzultace dopadl jinak,
- Zjemněním stupnice hodnot konstruktu z 5 základních hodnot na celou stupnici <-100, 100>, aby výsledek konzultace odpovídal představě experta.
- Dodáním hodnoty důležitost konstruktu ke každému konstruktu v %.

❑ Fáze využívání expertního systému

Je.li báze znalostí odladěna, to znamená, že se ve všech typech konzultací s expertem chová podle jeho představ, může být nabídnuta uživatelům.

Uživatel buď vybere konzultaci z již existujících, vytvořených expertem nebo některým předcházejícím uživatelem, a to podle názvu, cíle a zadání.

Pokud si nevybere, zadá svou novou konzultaci: zadá název a její zadání - požadavky, důležitost a hodnoty konstruktů. Uživatel vlastně zadává požadované hodnoty vlastností hledaného objektu, program vyhodnotí podobnost každého objektu tabulky se zadáním.

8.4. Příklad využití systému SAZZE

Teoreticky popsané etapy vývoje systému SAZE i jeho využití pro konzultace ukážeme na podrobně popsaném reálném příkladě [3]. Celý vývoj, ladění expertní báze i využití budeme prezentovat formou dialogu uživatele se systémem (**SYS**). V první fázi bude uživatelem expert (**EXP**), po odladění báze to bude běžný uživatel (**UŽIV**), požadující konzultace.

Příklad 10.5.

Báze znalostí o rekreačních oblastech.

Název: REKREACE
Cíl: Doporučení nejvhodnější rekreační oblasti požadovaných vlastností.
Objekty: Konkrétní rekreační oblasti.
Konstrukty: Umístění, roční doba, cena.

I. fáze - dialog při definování a naplňování repertoárové tabulky

1. Zadání objektů

SYS Zadejte konkrétní objekty (minimálně 5):

EXP Špindlerův Mlýn, Zemplínská Šírava, Balaton, Karlovy Vary, Benátky.

2. Zadání konstruktů

SYS Uveďte 1.významnou vlastnost, kterou dva z objektů Špindlerův Mlýn, Zemplínská Šírava, Balaton mají a jeden nemá:
EXP je na horách
SYS Jmenuje se opačný pól konstruktu takto? není na horách
EXP [potvrzení nebo oprava]
SYS Uveďte 2.významnou vlastnost, kterou dva z objektů Balaton, Zemplínská Šírava, Karlovy Vary mají a jeden nemá:
EXP je v cizině
SYS Jmenuje se opačný pól konstruktu takto? není v cizině
EXP [potvrzení nebo oprava]
SYS Uveďte 3. významnou vlastnost, kterou dva z objektů Balaton, Karlovy Vary, Benátky mají a jeden nemá:
EXP je u moře
SYS Jmenuje se opačný pól konstruktu takto? není u moře
EXP [potvrzení nebo oprava]

atd.

3. Zadání hodnot konstruktů pro všechny objekty
Po zadání objektů a konstruktů popisuje expert jednotlivé objekty.

SYS Popište každý objekt pomocí zadaných konstruktů.

Objekt Špindlerův Mlýn rozhodně je na horách
poněkud je na horách
nevím / ani jedno
poněkud není na horách
rozhodně není na horách

EXP rozhodně je na horách
atd.

Jednotlivé volby se transformují na hodnoty repertoárové tabulky. Výsledkem tohoto dialogu je repertoárová tabulka s hodnotami:

Objekt \ Konstrukt	je na horách	je v cizině	je u moře
Špindlerův Mlýn	100	-100	-100
Zemplínská Šírava	-100	100	-100
Balaton	-100	100	100
Karlovy Vary	50	-100	-100
Benátky	-100	100	100

Z každého pólu každého konstruktu zformuluje systém cílová pravidla. Příklad několika pravidel našeho příkladu:

SYS Zadaná pravidla jsou
Když je v cizině, pak
Zemplínská Šírava s váhou 100
Balaton s váhou 100
Benátky s váhou 100
Karlovy Vary s váhou -100
Špindlerův Mlýn s váhou -100
Když není v cizině. pak
Karlovy Vary s váhou 100
Špindlerův Mlýn s váhou 100
Zemplínská Šírava s váhou -100
Balaton s váhou -100
Benátky s váhou -100

ostatní pravidla pro jejich velké množství neuvádíme.

V rámci budování báze znalostí systém dále umožňuje provádět analýzu implikací a test shody objektů nebo konstruktů.

4. Analýza implikací

znamená nalezení všech dvojic logicky totožných implikací, které podle informací v bázi mohou platit. Analýzu provede systém a výsledky zobrazí. Expert pak potvrdí všechny skutečně platné. Ty jsou uloženy a budou využívány při konzultacích.

V našem příkladě se po provedení analýzy implikací objeví:

Analýza implikací

SYS Zdá se, že mezi vlastnostmi mohou platit následující vazby. Potvrďte platné z nich:
1. je na horách → není v cizině
je v cizině → není na horách
2. je na horách → není u moře
je u moře → není na horách
3. není v cizině → není u moře
je u moře → je v cizině

EXP [platné jsou implikace číslo 3, neplatné číslo 1]

SYS Znáte nějaký objekt, pro který neplatí implikace číslo 1?
EXP Insbruck
[zadá objekt ... a k němu hodnoty všech konstruktù], výsledkem je tabulka:

Objekt \ Konstrukt	je na horách	je v cizině	je u moře
Špindlerův Mlýn	100	-100	-100
Zemplínská Šírava	-100	100	-100
Balaton	-100	100	100
Karlovy Vary	50	-100	-100
Benátky	-100	100	100
Insbruck	100	100	-100

Expertem potvrzené implikace tvoří mezilehlá pravidla expertního systému.

5. Shoda objektù

znamená nalezení objektù, které mají ve všech konstruktech stejné hodnocení, porovnání se provádí procentuelně. Ve výsledku se uvádějí shody dvojic objektů nad 80%. Expert může zadat nový konstrukt, který oba objekty rozliší.

Procentuelní shoda dvou objektů Oi a Oj (z celkového počtu n objektů popsaných m konstrukty) se vypočítá podle vztahu

$$S_{ij} = \sum_{k=1}^{n} ((x_{ik} - x_{jk}) / 200) * 100 / m$$

kde x_{ik} je hodnota konstruktu k u objektu Oi.

Shoda objektů

SYS Podobnost mezi objekty Špindlerův Mlýn a Karlovy Vary je 92%. Znáte nějakou vlastnost, která by je rozlišila?
EXP možnost zimních sportů
SYS Jmenuje se opačný pól konstruktu takto? nemožnost zimních sportů
atd.

Následuje doplnění hodnot nového konstruktu u všech objektů, výsledkem je tabulka:

Objekt \ Konstrukt	je na horách	je v cizině	je u moře	zimní sporty
Špindlerův Mlýn	100	-100	-100	100
Zemplínská Šírava	-100	100	-100	-100
Balaton	-100	100	100	-100
Karlovy Vary	50	-100	-100	0
Benátky	-100	100	100	-100
Insbruck	100	100	-100	100

6. Shoda konstruktů

Obdobně se mohou objevit podobné póly některých konstruktů, **shoda konstruktů**. Řešení je obdobné, jako u shody objektů, za hranici se bere 70%.

Shoda konstruktů

SYS Podobnost mezi konstrukty je na horách a možnost zimních sportů je 83%. Pokuste se najít objekt, který by měl zároveň vlastnosti
je na horách a nemožnost zimních sportů nebo
není na horách a možnost zimních sportů.
EXP neznám
[zadá nový objekt a k němu hodnoty všech konstruktù, nebo takový objekt není znám]

Sloučení konstruktů: pokud je shoda konstruktů dána nevhodně zvolenou skladbou konstruktů, je vhodné tyto konstrukty sloučit do jednoho. Jeho jméno mùže být shodné s jedním z původních nebo je možno zadat nové.

SYS Mají se tedy konstrukty je na horách a možnost zimních sportů spojit do jediné vlastnosti?
EXP ano
SYS Jak se bude nový konstrukt jmenovat:
je na horách / není na horách
možnost zimních sportů / nemožnost zimních sportů
jinak
zrušit oba konstrukty
EXP jinak : vhodný pro zimní rekreaci
atd.,

systém se ptá na název opačného pólu, zruší 2 původní konstrukty, vytvoří nový, expert zadá hodnoty nového konstruktu pro všechny objekty (případnou modifikací hodnot zrušených konstruktů).

Výsledkem je tabulka:

Objekt \ Konstrukt		je v cizině	je u moře	zimní rekreace
Špindlerův Mlýn		-100	-100	100
Zemplínská Šírava		100	-100	-100
Balaton		100	100	-100
Karlovy Vary		-100	-100	-50
Benátky		100	100	-50
Insbruck		100	-100	100

7. Konzultace s expertem

Po vytvoření báze znalostí expert konzultacemi ověřuje, jak se shodují doporučení expertního systému s jeho vlastními a v případě neshody modifikuje bázi tak dlouho, dokud systém nedává odpovědi stejné, jako on.

Každá konzultace má svůj název a je uloženo její zadání - požadavky, důležitost a hodnota konstruktů. Je možno vybrat z již existujících nebo definovat novou.

Konzultace s expertem

SYS Vyberte si konzultaci:
Vánoce a Nový rok
nová
EXP nová: Letní rodinná s dětmi
SYS [pro všechny konstrukty se zadává jeho důležitost pro danou konzultaci]
Jak důležitý je konstrukt Je v cizině / není v cizině
velmi
trochu
vůbec ne
EXP trochu
atd. pro všechny konstrukty.
SYS U těchto konstruktů jste zadal,že pro konzultaci nejsou důležité:
vhodné pro zimní sporty
Pokud tomu tak není, doplňte zadání konzultace.
EXP [případné doplnění zadání nebo potvrzení platnosti zadání]
SYS [dále se systém ptá na požadované hodnoty konstruktů označených za velmi nebo trochu důležité]
EXP [zadá požadované hodnoty]

Pro náš příklad je celkem zadáno:

Je trochu důležité, aby platilo rozhodně není v cizině.
Je trochu důležité, aby platilo rozhodně není u moře.
Je velmi důležité, aby platilo rozhodně možnost zimních sportů.
Je velmi důležité, aby platilo poněkud laciné.

Uživatel vlastně zadává požadované hodnoty vlastností hledaného objektu, program vyhodnotí podobnost každého objektu tabulky se zadáním.

Po ukončení zadání se zobrazí výsledek konzultace ve tvaru:

SYS Pro konzultaci Rodinná s dětmi systém doporučuje objekty
83 % Zemplínská Šírava
55 % Balaton
27% Karlovy Vary
Systém nedoporučuje
14 % Špindlerův Mlýn
-14 % Benátky
-46 % Insbruck

[doporučuje objekty s váhou > 20, nedoporučuje ostatní]

Pokud s některým zařazením nesouhlasíte, označte příslušný objekt.

8. Zlepšování chování báze

- Přidáváním nových objektů
  - podle nápovědy, dotazem na nový objekt s hodnotami

    konstruktů, jejichž kombinace v bázi nejsou,
  - bez nápovědy
- Přidáváním nových konstruktů

  - bez nápovědy
  - podle nápovědy využitím mezilehlých pravidel (výsledku analýzy implikací), pro neplatná doplnit konstrukty s hodnotou různou u doporučených objektů a u nedoporučených objektů,
  - podle nápovědy s využitím výsledku konzultace, dotazem na nový konstrukt s hodnotami různými pro expertem doporučené a nedoporučené objekty, přičemž výsledek konzultace dopadl jinak,
- Zjemněním stupnice hodnot konstruktu z 5 základních hodnot na celou stupnici <-100, 100>, aby výsledek konzultace odpovídal představě experta.
- Dodáním hodnoty důležitost konstruktu ke každému konstruktu v %.

Výsledná upřesněná a doplněná repertoárová tabulka našeho příkladu:

Objekt \ Konstrukt	v cizině	u moře	zimní rekreace	vodní sporty	drahá	stanování
Špindlerův Mlýn	-100	-100	100	-100	-100	-100
Zemplínská Šírava	100	-100	-100	100	-100	100
Balaton	100	-100	-100	100	50	100
Karlovy Vary	-100	-100	-50	-50	50	-100
Benátky	100	100	-50	100	100	100
Insbruck	100	-100	100	-50	100	-100

II. Konzultace uživatele s expertním systémem

Uživatel expertního systému konzultuje s hotovým naplněným systémem obdobným způsobem, jako expert při ověřování báze znalostí, pouze bez možnosti změn báze.

Výsledkem je výpis konzultace obsahující název konzultace, zadání a výsledná doporučení s váhou.

Mimo to může uživatel požádat o výpisy

- pravidel cílových i mezilehlých (výsledků expertem uznaných implikací),
- báze znalostí (repertoárové tabulky),
- výsledku uložených konzultací.

8.5. Implementace systému SAZZE

Na katedře informatiky VŠB-TUO byl v rámci diplomové práce [15] implementován programový systém SAZZE, který realizuje všechny popsané funkce. Ukážeme si jeho ovládání opět na příkladu z praxe.

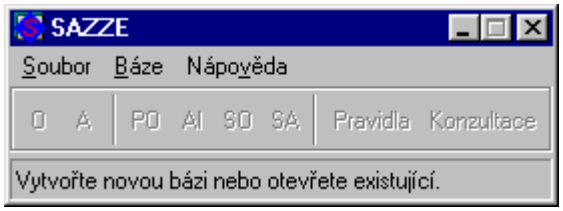
Příklad 10.6.

Vytápění

S pomocí experta z oblasti návrhu druhu a způsobu vytápění bude ukázkově sestavována báze znalostí expertního systému, který bude schopen z daného hlediska doporučit druh topení. Vytápění se týká běžných objektů pro bydlení, tj. objektů, které mají topení s výkonem kotle do 50 kW. Není zde zahrnuto vytápění velkých budov (panelové domy, školy, úřady, atd.) a dálkové

způsoby vytápění (dostupnost pouze v nevelké vzdálenosti od poskytovatelů odpadního tepla). Netýká se ani výrobních objektů.

Po spuštění aplikace SAZZE se zobrazí následující okno:

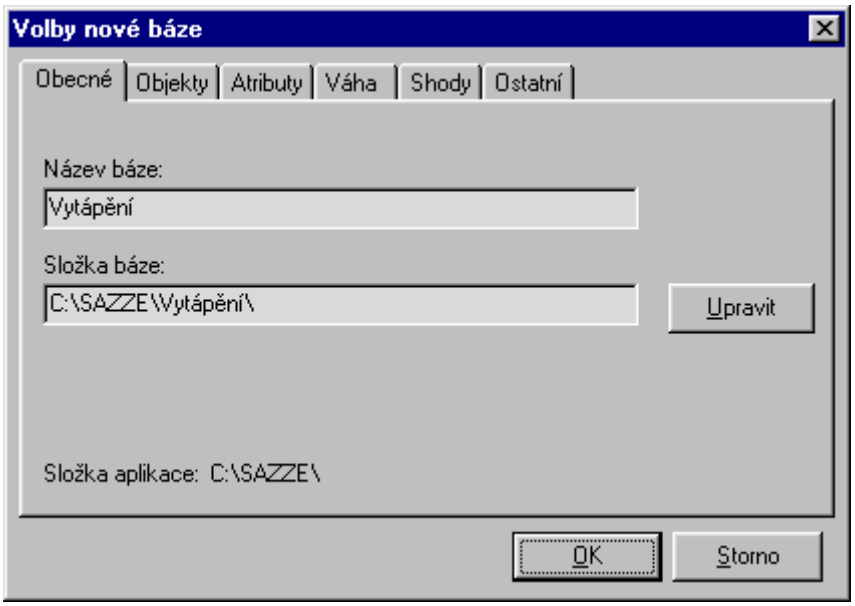


Na nástrojové liště nyní není dostupná žádná ikona (tak jako v menu Báze žádná položka), protože není otevřena žádná báze dat a znalostí. Jelikož vytváříme novou bázi, nebudeme otevírat žádnou již existující bázi.

Než začneme analyzovat problém vytápění, vytvoříme tedy prázdnou bázi výběrem položky Nová v hlavním menu aplikace Soubor. Zobrazí se formulář, do kterého vyplníme v záložce Obecné název nové báze „Vytápění“, v záložce Shody nastavíme hranici uvádění shod objektů na 90 % a v záložce Ostatní vyplníme jméno experta. Ostatní údaje použijeme standardní.

Navržené objekty, jejich konstrukty a váhy jsou zřejmé z repertoárové tabulky, pro stručnost tedy neuvádíme kompletní dialog.

Aby se dala určit hranice, kdy je instalace (nebo provoz) daného objektu drahá a kdy levná, bereme za rozhodně drahou nejdražší instalaci a za rozhodně levnou nejlevnější instalaci.



Většina objektů využívá při provozu také jiný druh energie, i když její spotřeba není významná. Výjimku tvoří objekty č. 6 – Solární energie a č. 7 – Tepelné čerpadlo, které by bez elektrické energie nemohly z technologického hlediska správně fungovat. Při popisu těchto (i ostatních) objektů byl na tento fakt brán zřetel a váha atributu určujícího finanční náročnost provozu upravena.

Do atributu „je ekologické“ byla zahrnuta také obnovitelnost a neobnovitelnost zdrojů energií. Dřevo jako obnovitelný zdroj energie má větší váhu než elektrická energie. Pro ni, ačkoliv je obnovitelným zdrojem, se využívají neobnovitelné zdroje, jako je uhlí a plyn.

Atribut „je rychlý ohřev topného média“ byl zvolen kvůli potřebě energie při přerušovaném provozu, kde je spotřeba energie ovlivňována akumulační schopností budov. Je-li schopnost budovy akumulovat energii nízká, je výhodná pro přerušovaný provoz oproti stejně zateplené budově ale s větší akumulací tepla.

Vytváření báze znalostí:

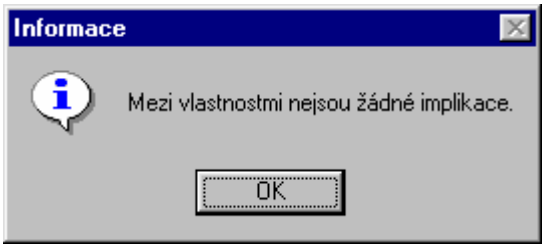
1. *Zadáme všechny objekty.*
2. *Pokračujeme zadáním navržených atributů.*
3. *Pomocí formuláře pro popis objektů vytvoříme repertoárovou tabulku. Použijeme jemnou stupnici, takže můžeme také zadávat hodnoty jiné, než na pětibodové stupnici. Ohodnocení atributů je následující:*

Repertoárová tabulka

	instalace je drahá	provoz je drahý	je ekologické	rychlost ohřevu
dřevo	-100	-50	70	60
elektřina	30	100	50	100
hnědé uhlí	-50	-20	-50	40
kaly	-50	-50	-100	20
koks	-50	10	20	10
solární energie	80	-80	80	-90
tepelné čerpadlo	100	-100	100	-50
zemní plyn	60	40	70	80
černé uhlí	-50	-10	-30	50

4. *Provedeme analýzu implikací (AI).*

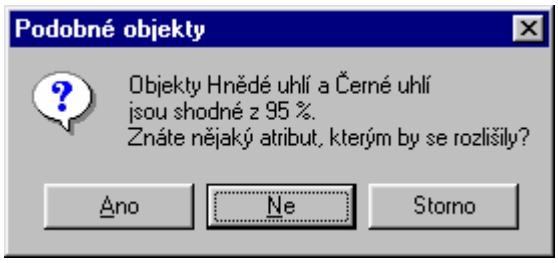
V našem případě nebyla nalezena žádná implikace:



5. *Analýza SO, SA*

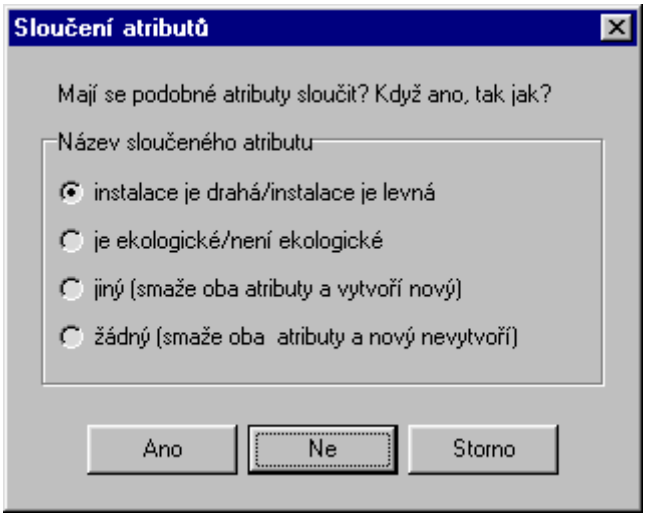
- *Po provedené analýze shod objektů se objeví okno s informací, že objekty „hnědé uhlí“ a „černé uhlí“ jsou shodné z více než 90 %, a to z celých 95 %. Protože tyto zdroje energie jsou z principu podobné (přibližně stejně drahé, stejně dostupné a mají podobný způsob spalování), nebudeme vkládat žádný nový atribut, kterým by se rozlišily.*

  Na následující dotaz odpovíme stisknutím tlačítka Ne.



- *Následně provedeme analýzu shod atributů. Je zobrazeno okno s informací, že atributy „instalace je drahá“ a „je ekologické“ jsou shodné z 81 %, a že atributy „provoz je drahý“ a „ohřev topného média je rychlý“ jsou shodné ze 78 %. Po výběru první možnosti je po nás požadováno vložení nového objektu uvedených vlastností. Protože platí, že co je ekologické bývá i drahé, nebudeme shodu těchto atributů vyvracet.*

Na dotaz odpovíme ne a zobrazí se okno s výběrem způsobu sloučení těchto podobných atributů:Protože atributy nebudeme slučovat ani mazat, odpovíme rovněž ne. Touto poslední odpovědí jsme získali přístup ke zbývajícím dvěma položkám lišty, kterými jsou Pravidla a Konzultace.

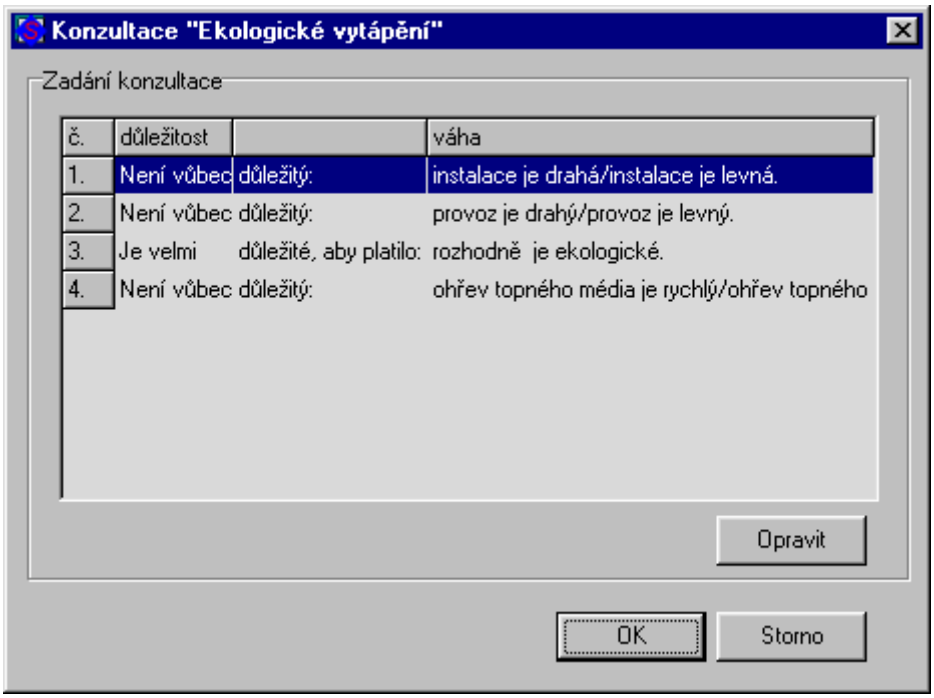


6. *Pravidla platící v bázi jsou buď cílová nebo mezilehlá. Mezilehlá pravidla v naši bázi nejsou, protože zde neplatí žádná implikace. Protože pro ostatní atributy mají pravidla stejný tvar, stačí zde uvést příklad výpisu koncových pravidel pro jeden atribut. Koncová pravidla totiž vedou od každého pólu každého atributu ke všem objektům a jejich kompletní výpis by byl rozsáhlý.*

- Když je ekologické, pak:
  Tepelné čerpadlo s váhou 55.
  Solární energie s váhou 44.
  Dřevo s váhou 39.
  Zemní plyn s váhou 39.
  Elektřina s váhou 28.
  Koks s váhou 11.
  Černé uhlí s váhou -17.
  Hnědé uhlí s váhou -28.
  Kaly s váhou -55.
- Když není ekologické, pak:
  Tepelné čerpadlo s váhou -55.
  Solární energie s váhou -44.

Dřevo s váhou -39.
Zemní plyn s váhou -39.
Elektřina s váhou -28.
Koks s váhou -11.
Černé uhlí s váhou 17.
Hnědé uhlí s váhou 28.
Kaly s váhou 55.

7. *Nyní přistoupíme k testování báze znalostí – konzultacím. Po vložení názvu nové konzultace „Ekologické vytápění“ zadáme důležitosti a váhy jednotlivých atributů pro tuto konzultaci. Zadání je v následujícím formuláři:*



Pokud nás zajímá pouze ekologie a nevíme nic o nákladech na instalaci a provoz, považujeme za jediný velmi důležitý (100 %) atribut „je ekologické“. Jeho váhu zadáme „rozhodně ano“, tedy +100 (váha je v intervalu –100 až 100). Ostatní atributy nejsou vůbec důležité (0 %) a váhu tudíž zadávat nemusíme.

Výsledkem konzultace je následující okno, ve kterém jsou uvedeny všechny objekty s rozdělením do doporučujících intervalů a s uvedením váhy:

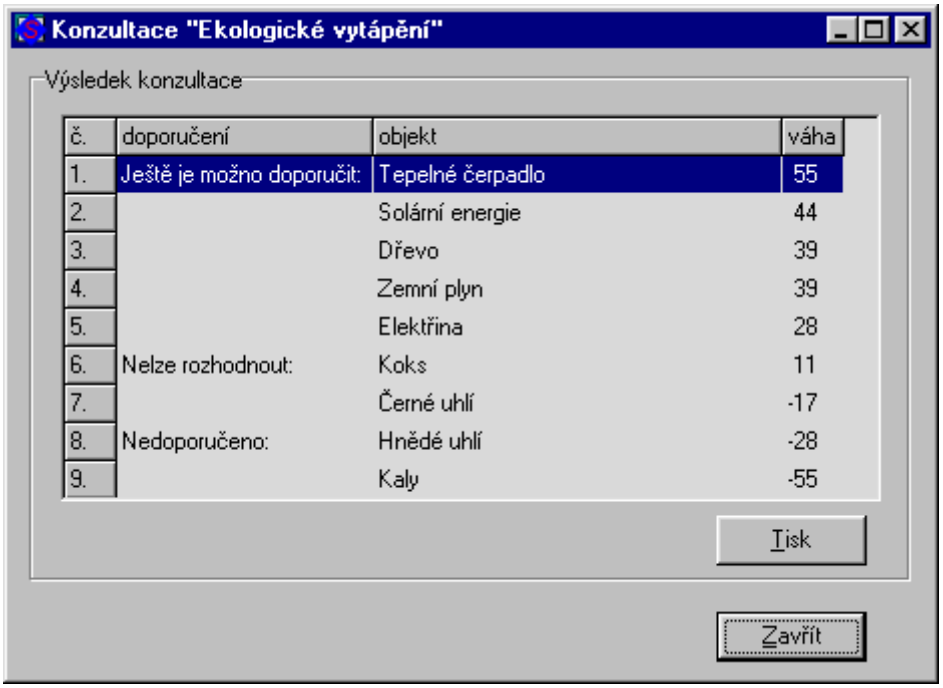
Konzultace

Jako příklad využití báze je uveden výpis konzultace, která doporučí co nejlevnější vytápění s ohledem na životní prostředí.

Konzultace "Levné vytápění"

A) ZADÁNÍ:

1. Je velmi důležité, aby platilo: rozhodně instalace je levná.
2. Je velmi důležité, aby platilo: rozhodně provoz je levný.
3. Je trochu důležité, aby platilo: spíše je ekologické.
4. Je trochu důležité, aby platilo: spíše ohřev topného média je rychlý.



Konzultace "Levné vytápění"

A) ZADÁNÍ:

1. Je velmi důležité, aby platilo: rozhodně instalace je levná.
2. Je velmi důležité, aby platilo: rozhodně provoz je levný.
3. Je trochu důležité, aby platilo: spíše je ekologické.
4. Je trochu důležité, aby platilo: spíše ohřev topného média je rychlý.

B) VÝSLEDKY *(váha doporučení je z intervalu <-100, 100>.*

Rozhodně doporučeno:
- Dřevo (váha 74)

Ještě je možno doporučit:
- Kaly (váha 41)
- Tepelné čerpadlo (váha 37)
- Hnědé uhlí (váha 35)
- Černé uhlí (váha 34)
- Koks (váha 27)

Nelze rozhodnout:
- Solární energie (váha 20)
- Zemní plyn (váha -13)

Nedoporučeno:
- Elektřina (váha -25)

Úlohy k řešení 10.

1. Zvolte si oblast, v níž se považujete za experta a vytvořte repertoárovou tabulku pro tuto oblast: navrhněte vhodné cíle využití běžnými uživateli, k nim vhodné objekty a konstrukty. Pro ně pak naplňte repertoárovou tabulku, případně ji několika testovacími pokusy odlaďte.

9. SW PRO PODPORU DOLOVÁNÍ ZNALOSTÍ

Čas ke studiu: 1 hodina

Cíl Po prostudování této kapitoly se seznámíte s

- SW produkty, které nabízejí některé z metod dolování znalostí
- metodami, které tyto SW produkty nabízejí.

Výklad

9.1. Obecně o SW pro Data Mining

Původně (historicky) byly implementovány jednotlivé metody samostatně a využívány většinou pro výzkumné účely nebo sociální průzkumy. Později – přibližně se vznikem datových skladů se začaly využívat metody získávání znalostí i pro shromážděná data a datové sklady. Výsledky přesahující prosté agregované dimenzionální řady a jejich hierarchie, jak již víme, metody mohou objevovat i dosud netušené souvislosti a pravidla v datech ukrytá.

Firmy nabízející Minery většinou nezveřejňují podrobnější informace o metodách nebo dokonce algoritmech, použitých ve svých produktech. Často popisují metodu jako „velmi podobnou“ metodě XYZ, ale vyvinutou vlastními pracovníky firmy. Zkušený programátor a analytik někdy odhadne i ze stručných informací metodu. Pokud však chce mít jistotu o výsledku metody, je nutné ji zřejmě otestovat na známých datech nebo porovnat se známými výsledky.

Často firmy také inzerují, že grafické uživatelské rozhraní a automatizovaný rámec znamenají, že nemusíte vědět, jak nástroje dolování pracují k tomu, abyste je používali. Tvrdí například, že „obchodní technolog s malou statistickou odbornou znalostí může rychle a snadno používat systém“. Není to pravda. Je sice často pravdou, že uživatel bez znalostí metod může snadno a rychle naklikat zadání a dostat nějaké výsledky. Pokud ale neví, co vlastně výsledky znamenají, jak se mohou interpretovat, nejsou mu buď k ničemu, nebo – což je horší – si je může interpretovat chybně a nadělat tak i škodu. Prakticky by měli s nástroji pro dolování dat pracovat (na žádost managementu nebo při vlastní analytické práci) jen analytici. I těm snadné a rychlé ovládání systému velmi urychlí práci.

Následující stručné informace jsou převzaty z reklamních materiálů firem nabízejících jednotlivé produkty. Proto je každý z nich „nejlepší nebo jediný“. Proto jsou zde někdy přidány k takovým tvrzením uvozovky. Pravidelně systémy pro dolování dat nabízí i základní metody matematické statistiky. Často je uvádí dohromady s metodami dolování, proto i v našem přehledu budou uváděny tak, jak jsou prezentovány

Všechny popisované systémy se ale stále vyvíjejí a jsou nabízeny jejich nové a nové verze. Proto se nespoléhejte na níže uvedené informace a před praktickým použitím kteréhokoliv DM systému prostudujte aktuální manuál.

Následující systémy můžeme rozdělit do dvou skupin.

První tvoří profesionální komerční SW systémy, s dokonalým uživatelským prostředím, někdy poněkud neprůhlednými metodami, většinou velmi drahé. Mívají robusní podporu školeními (také velmi drahými) a jsou tak použitelné především velkými firmami s potřebou analyzovat svá data. Patří mezi ně například

Enterprise Miner firmy SAS Institute	http://www.sas.com
Clementine firmy SPSS	http://www.spss.com
Intelligent Miner firy IBM	http://www-4.ibm.com
Statistice Data Miner firmyStatSoft	http://www.statsoft.com
CART firmy Salford Systém	http://www.salford-system.com

Data Miningové rozšíření mají i velké databázové systémy, například

Oracle Data Miner firmy Oracle	http://www.oracle.com
SQL Server 20xx firmy Microsoft	http://www.microsoft.com

Druhou skupinu tvoří volně dostupné SW systémy, většinou vyvíjené na některé univerzitě. Nemají vždy pokryty všechny metody, ale zase někdy mají velmi dobré implementace algoritmů. Postupně se sobě stále více podobají z hlediska ovládání metod.

V následujících kapitolkách představíme stručně některé z nich. Podrobněji jeden open-source **Rapid Miner** je popsán podrobněji v samostatné příloze.

9.2. SPSS Clementine

Clementine je produkt firmy Integral Solutions. V polovině 90. let začala vyvíjet tento software, v roce 1999 ji odkoupila firma SPSS, kterou v roce 2010 odkoupila další firma IBM. Ta tento produkt přejmenovala z Clementine na IBM SPSS Modeler.

Informace získáme na adrese http://www.spss.com

Clementine využívá metodologii CRISP-DM. Systém je komerční a na webových stránkách autora není demoverze ke stáhnutí. Vyskytuje se pouze v těchto verzích:

- IBM SPSS Modeler Professional
- IBM SPSS Modeler Premium
- IBM SPSS Modeler Server

Popisovaný systém je Clementine ve verzi 12.0.

❑ Funkce

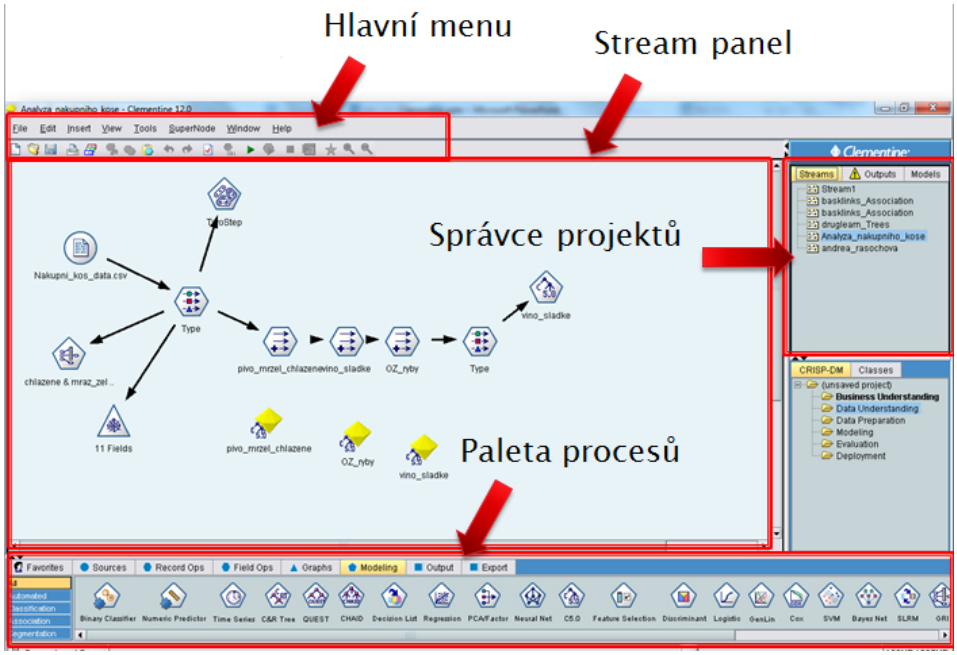
Systém nabízí zpracování mnoha data miningových metod a algoritmů. Mezi nejdůležitější z nich bych zařadil tyto:

- Asociace
- Rozhodovací stromy (algoritmy C&RT, Chaid, Quest, ...)
- Neuronové sítě
- Regresní algoritmy
- další ...

Clementine je vhodný pro zpracování objemnějších datových souborů a garantuje vyšší přesnost modelů díky implementaci pokročilejších algoritmů (Bagging, Boosting, ...).

Pracovní prostředí

Pracovní prostředí, které je vidět na obrázku 1, je rozděleno na hlavní menu, které je umístěno v horní části. Zde lze najít i nápovědu programu a nastavení. Spouští se zde i kompletní navrhnutý stream. Pod hlavním menu je Stream panel, který slouží k poskládání jendotlivých bloků tak, aby plnily námi navrhnuté funkce. Jednotlivé bloky, které lze na Stream panel přidávat najdeme ve spodní paletě procesů, kde pod jednotlivými záložkami najdeme procesy roztříděné. V pravé části programu se nachází správce projektů a vypočítaných modelů.



Obrázek 9.1: Pracovní plocha Clementine

Paleta procesů

Na spodní paletě najdeme pod záložkami skupiny procesů roztříděných podle kategorií. Jednotlivé kategorie si dále popíšeme.

Sources

Na obrázku 2 je vidět výčet procesů patřících do této kategorie. Jedná se o proces, který načítá vstupní data ze zadaných zdrojů. Může to být například databáze, Excel soubor, CSV, apod.

Obrázek 9.2: Sources

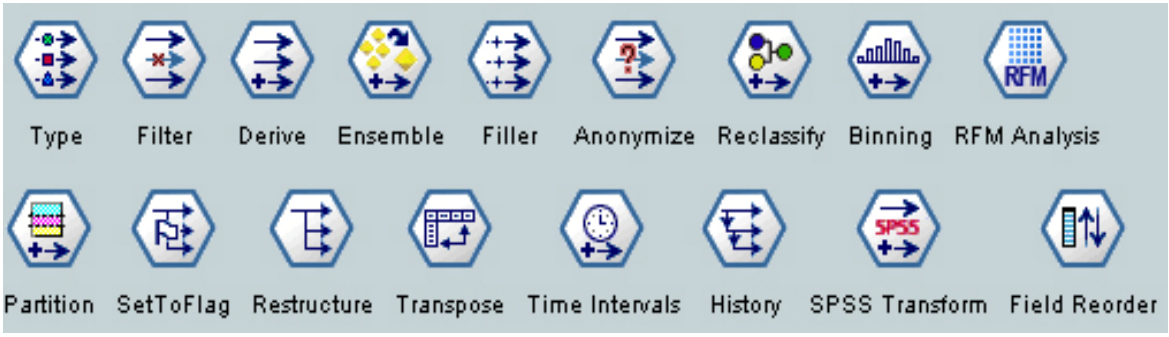
Record Ops

Tato záložka obsahuje procesy pro prvotní operace s daty, jak je sloučení dat, výběr určitého vzorku dat apod. Příklad těchto procesů je na obrázku 3.

Obrázek 9.3: Record Ops

Field Ops

Tyto procesy mají za úkol zpracovávat určitým způsobem již vybraná a načtená data ze zdroje. Můžou data filtrovat, vybírat data a třídit je před vstupem do jednotlivých metod algoritmů. Příklad je vidět na obrázku 4.



Obrázek 9.4: Field Ops

Graphs

Procesy Graphs poskytují reprezentaci výsledků analýz ve formě grafického zobrazení pomocí grafů. Příklad těchto procesů je vidět na obrázku 5

Obrázek 1.5: Graphs

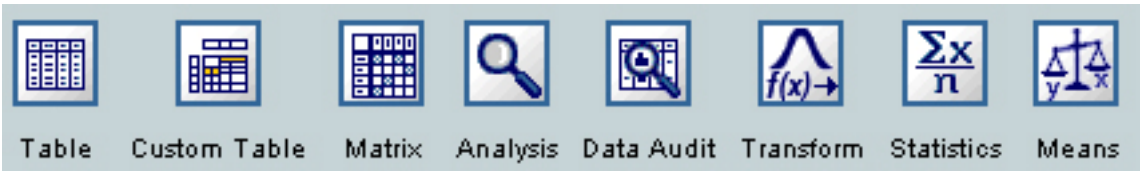
Modeling

Na obrázku 6 je paleta procesů, která je již určená pro nasazení jednotlivých metod data miningu na naše data. Výčet těchto metod není konečný, program jich obsahuje daleko více. Je zde vidět metoda K-středů, Apriory algoritmus, GRI apod.

Obrázek 9.6: Modeling

Output

Procesy na obrázku 7 slouží pro uložení výstupu výsledků naší analýzy. Možné je uložit výsledky do tabulky, matice, jiné analýzy apod.



Obrázek 9.7. Output

Export

Export slouží k exportování výsledků do jiných souborů, jako jsou Excel apod., nebo například do databáze. Příklad těchto procesů je vidět na obrázku 9.

Obrázek 9.8: Export

9.3. WEKA

Úvod a historie

Systém Weka reprezentuje kategorii nekomerčních, volně dostupných systémů vyvíjených v akademickém prostředí. Je vyvinutý na universitě Waikato na Novém Zélandě (Witten, Frank, 1999). Původním záměrem bylo vytvořit kolekci technik data miningu pro posílení novozélandské ekonomiky, zejména zemědělství. I přes konkurenci v podobě RapidMineru se projektu stále daří. Weka má charakter experimentálního systému nabízejícího širokou škálu algoritmů. K dispozici jsou i možnosti vizualizace a kombinování modelů. WEKA byla postupně přepsána do Javy a je publikována pod svobodnou licencí GPL.

Stažení

Aplikace je dostupná na internetu na adrese

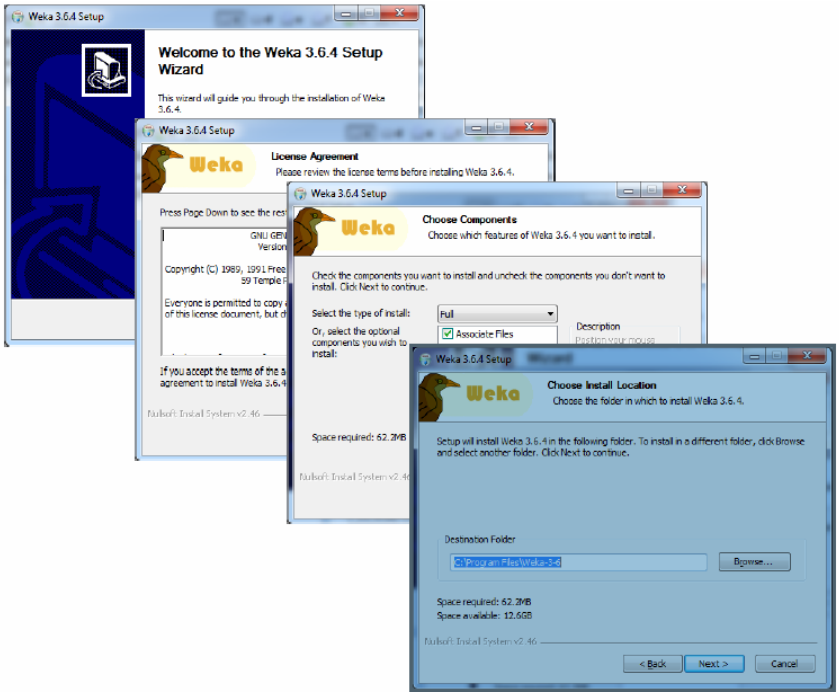
http://www.cs.waikato.ac.nz/ml/weka/

V levém menu zvolíme *Download* a vybereme potřebnou verzi systému (MS Windows, Linux, Mac OS X). Je třeba ale vybírat v *Stable GUI vision*, stabilní verzi. Vzhledem k tomu, že je Weka implementována v javě, tak pro její spuštění je tedy třeba mít nainstalován Java VM. Ke stažení jsou verze s Javou i bez ní.

Popisována je verze 3.6.4 soubor *weka-3-6-4-x64.exe.*

Instalace

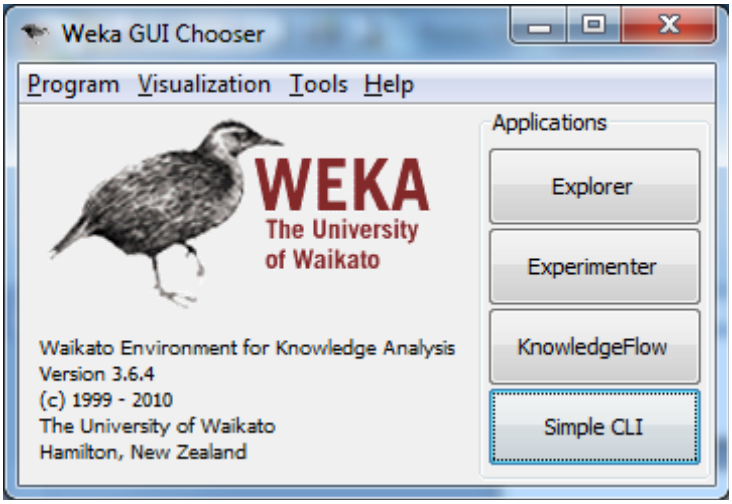
Instalace je jednoduchá pomocí klasického průvodce.



Obrázek 9.9: Spuštění programu Weka a výběr GUI

Spuštění

Program lze spustit buď přes ikonu v hlavní nabídce a můžeme si vybrat klasické GUI rozhraní anebo pomocí příkazové řádky, nebo také lze program spustit přímo poklepáním na soubor *weka.jar*, v místě kde je nainstalovaný. Zjistil jsem, že je možno spustit program i pomocí *RunWeka.bat*.



Obrázek 9.10: Spuštění programu Weka a výběr GUI

❑ Použití

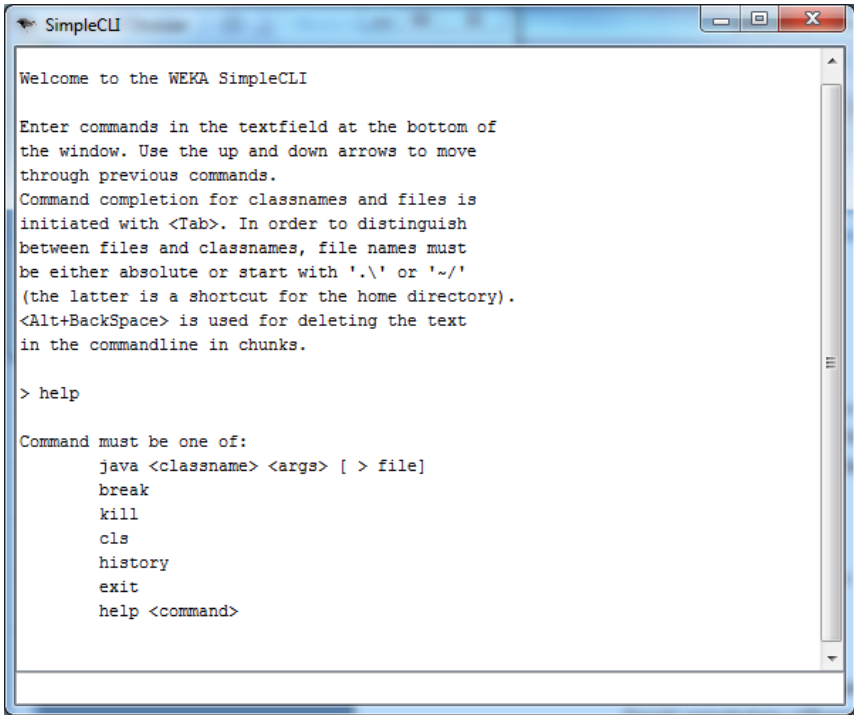
Weka má čtyři různá GUI, lze s ní pracovat i přes příkazový řádek, nebo ji využít jako knihovnu ve vlastním kódu.

Druhy GUI

- Simpli CLI
- Knowledge Flow
- Experimenter
- Explorer

Simpli CLI

Nejedná se o plnohodnotné GUI, ale spíše jen o konzoly v grafickém okně. Oproti samotnému příkazovému řádku nabízí jen několik vylepšení. Není třeba se zabývat CLASPATH, je zde funkce doplňování příkazů a doplňování jmen souborů. Moc se nevyužívá, snad na rychlé vyzkoušení.

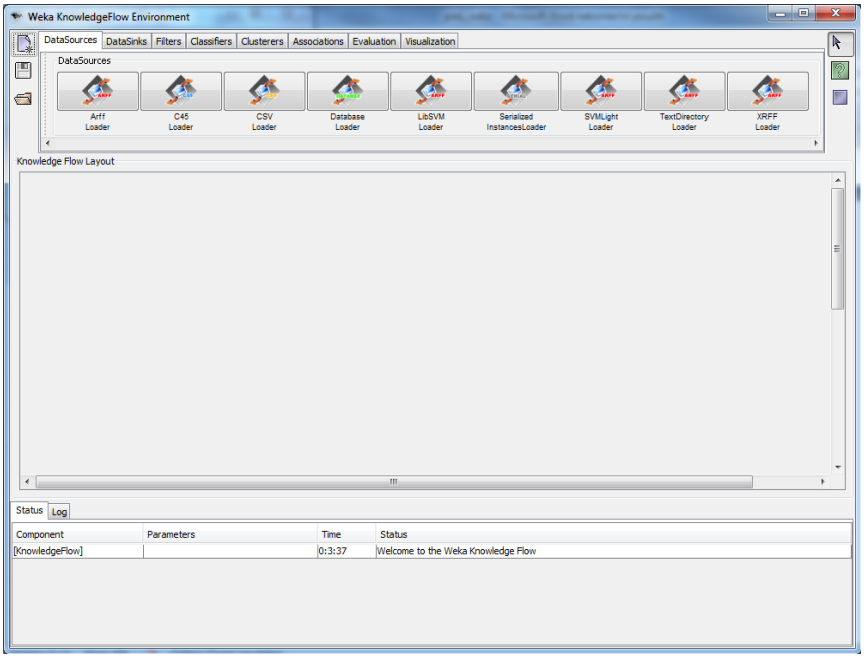


Obrázek 9.11: Simpli CLI - GUI prostředí

Knowledge Flow

Používá rozhraní, které je podobné RapidMineru, tzv. data-flow přístup. Obsahuje několik typů objektů. Některé slouží k načítání či ukládání dat, dalšími jsou samotné filtry, klasifikátory a shlukovací algoritmy. Jinými objekty jsou různé metanástroje, které slouží např. k vizualizaci, hodnocení výkonnosti apod. Objekty se naskládají na hlavní plochu a napojují se na sebe (např. výstup komponenty ArffLoader může být vstupem nějakého shlukovacího algoritmu). Toto GUI má několik významných výhod, mezi nejvýznamnější z nich patří:

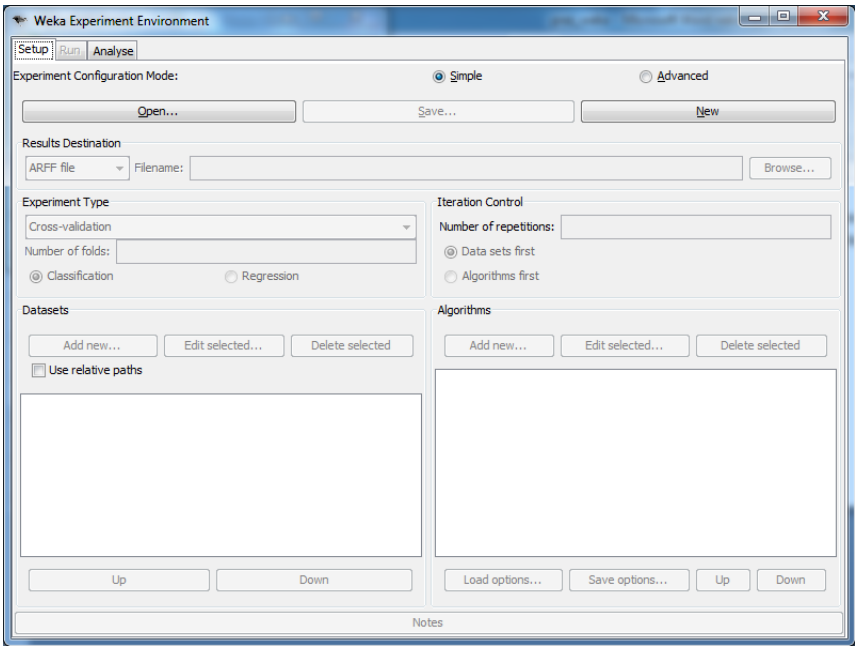
- možnost zpracovávat trénovací data jak dávkově, tak po jednotlivých vektorech
- řetězení různých filtrů, které velmi usnadní předzpracování dat
- paralelní zpracovávání pomocí vláken
- vizualizace, která probíhá zároveň se zpracováním dat



Obrázek 9.12: Knowledge Flow - GUI prostředí

Experimenter

Experimenter slouží k jednoduchému provádění opakovaných experimentů. Používá se mimojiné pokud je naším cílem testovat výkonnost několika algoritmů na stejných datech.



Obrázek 9.13: Experimenter - GUI prostředí

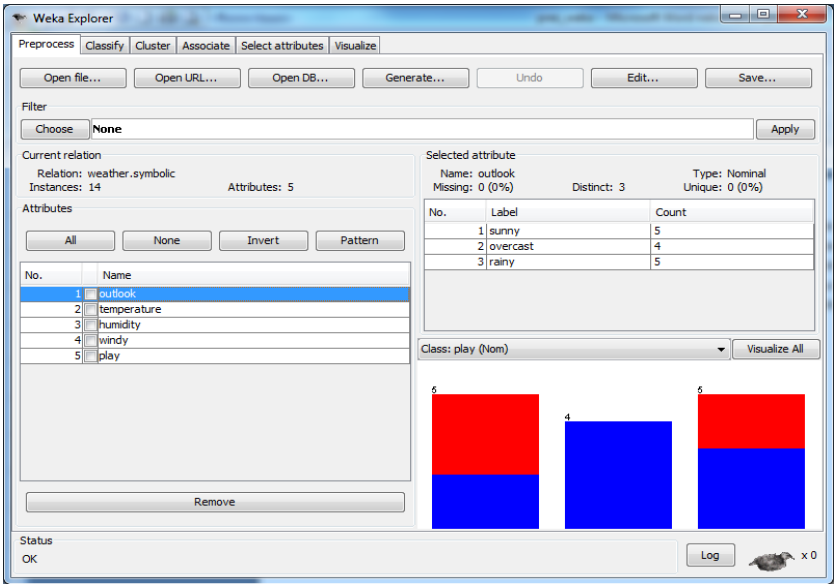
Explorer

Načtení

Toto GUI je nejpoužívanější. Hlavní okno je rozděleno do několika panelů, ale aby se nám všechny panely zpřístupnily, je třeba načíst data, se kterými budeme pracovat. V panelu *Preprocess* můžeme otevřít data buď pomocí databáze, URL adresy či souboru a to několika typů. Vlastní formát souboru

*.arff, kromě toho ještě typ C4.5 *.names, *.data; *.xrff, *.csv a další. Po nainstalování je v aplikaci několik jednoduchých dat, které jsou k dispozici na vyzkoušení některých algoritmů a funkcí aplikace Weka.

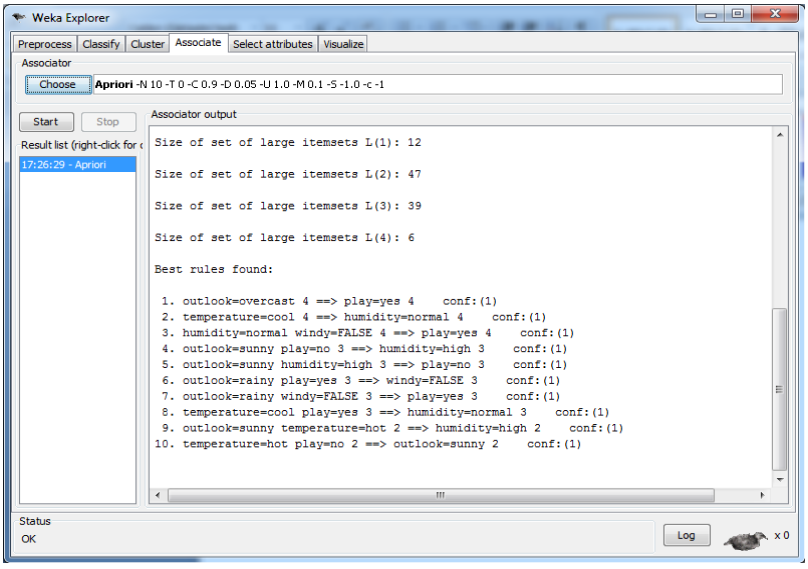
Po otevření dat se objeví seznam atributů a lze vidět několik statistik. Pokud vybereme některý atribut, tak vidíme četnost jednotlivých hodnot, v případě, že se jedná o kategoriální proměnnou; minimum, maximum, průměrná hodnota či směrodatná odchylka, v případě, že se jedná o numerickou hodnotu. Vedle toho je i histogram. V poslední řadě ještě musím zmínit část pro před filtraci a předzpracování dat (upravení - editace) a nastavení jeho parametrů.



Obrázek 9.14: Explorer - GUI prostředí, načtení dat a statistiky

Hledání asociací

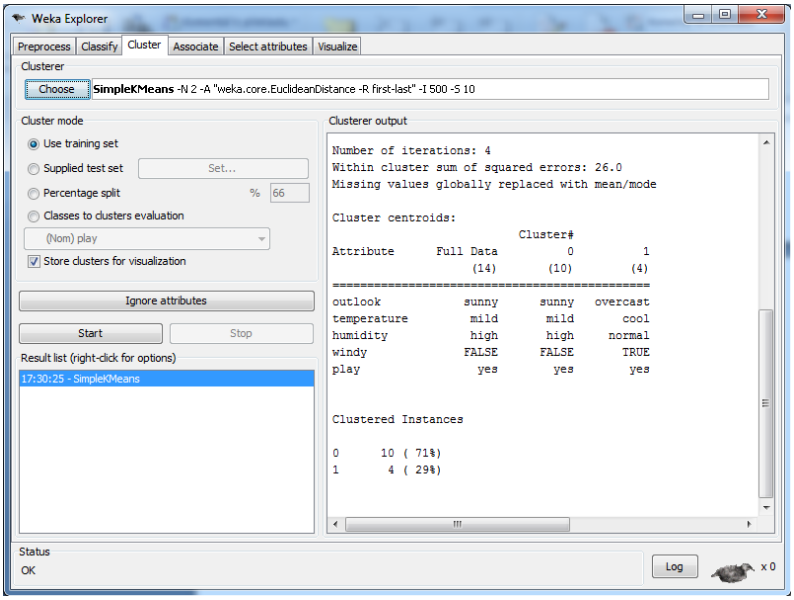
Přepneme na záložku *Asocciate* a použijeme načtená data. Na ukázku si zobrazíme známý algoritmus *Apriori*. Ten vybereme v menu *Choose*, kde jsou vidět další algoritmy. Za názvem algoritmu je řada dalších nastavení parametrů a můžeme se dočíst o informacích algoritmu a významu parametrů. Výsledkem je seznam asociačních pravidel.



Obrázek 9.15: Explorer - Asociace

Shlukování

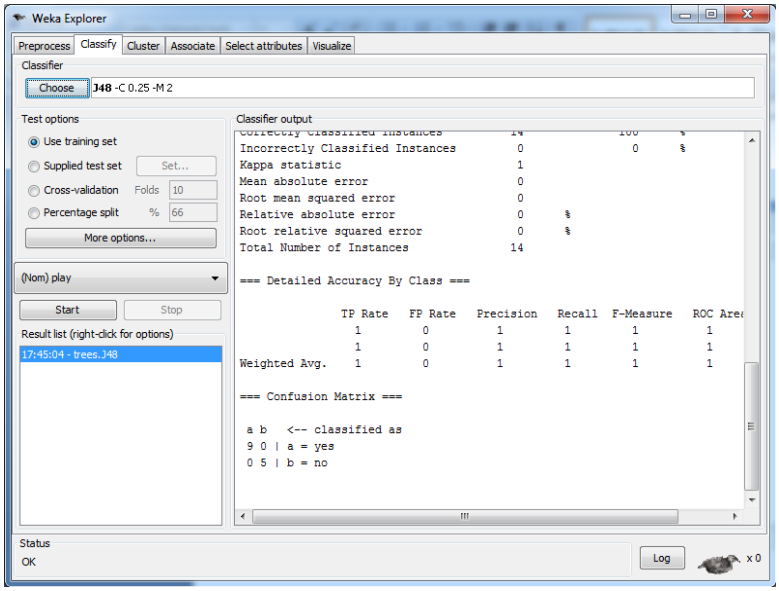
Přepneme se na jiný panel s názvem *Cluster*, který slouží ke shlukování. Je zde opět pár nastavení parametrů. Také se zde dá dočíst o základních informacích algoritmu a významu parametrů. Shlukování spustíme tlačítkem *Start*.



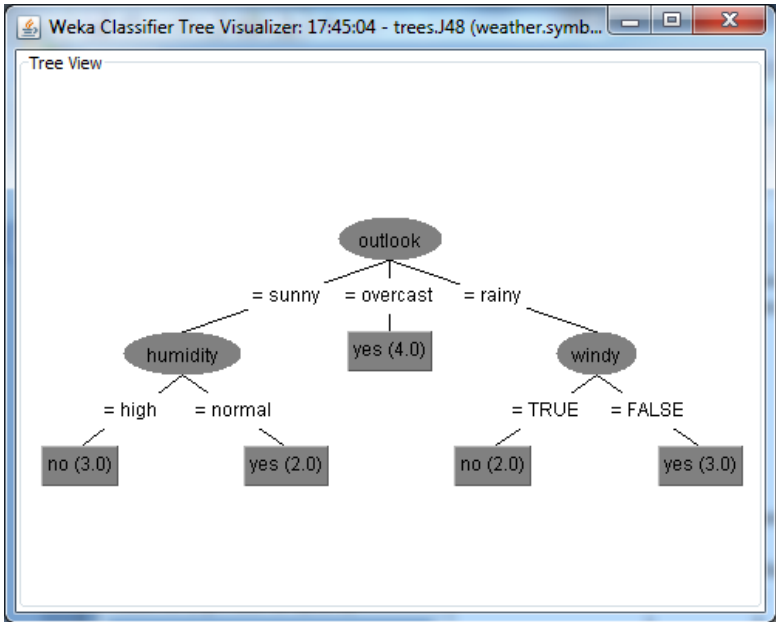
Obrázek 9.16: Explorer – shlukování

Klasifikační úlohy – stromy

Pro klasifikaci je zde použit algoritmus *J48*. Parametry pro nastavení: kromě základního tlačítka pro výběr klasifikátoru si můžeme vybrat mezi několika způsoby testování. Několik dalších voleb lze nastavit pomocí tlačítka *More options*. První parametr rozhoduje o prořezání stromu a druhý parametr rozhoduje o minimálním počtu prvků v listu. Spustíme tlačítkem *Start.*



Obrázek 9.17: Explorer – klasifikace



Obrázek 9.18: Explorer - klasicikace a ukázka stromů

9.4. SUMATRA

Úvod a historie

Sumatra TT je originální universální systém, vyvinutý na ČVUT v Praze, určený pro předzpracování dat. Dovoluje přistupovat a transformovat data z nejrůznějších typů zdrojů (např. plain-text, SQL, DBF, XML atd.).

Data jsou zpracovávána pomocí sady nabízených modulů obsahujících matematické výpočty, skriptování, XSLT a specializované funkce, nebo je dále možné vizualizovat zpracovaná data pomocí grafů, histogramů, 3D grafiky, nebo jako XML stromy.

Stažení

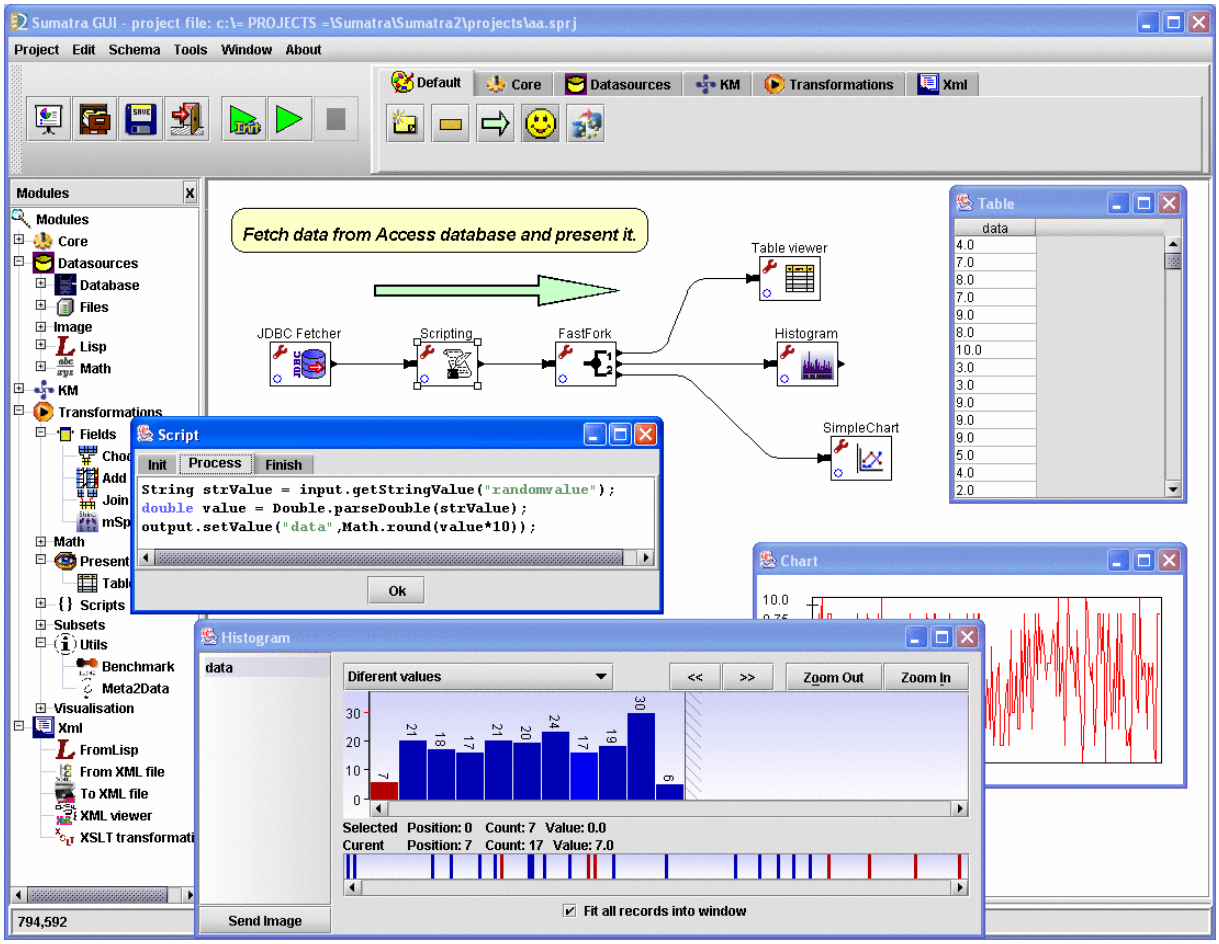
Systém je možné získat zdarma z těchto oficiálních stránek:

http://krizik.felk.cvut.cz/sumatra/index.html

Zde je odkaz přímo na archiv obsahující veškeré soubory potřebné ke spuštění:

http://krizik.felk.cvut.cz/sumatra/SumatraTT-2.1.1.zip

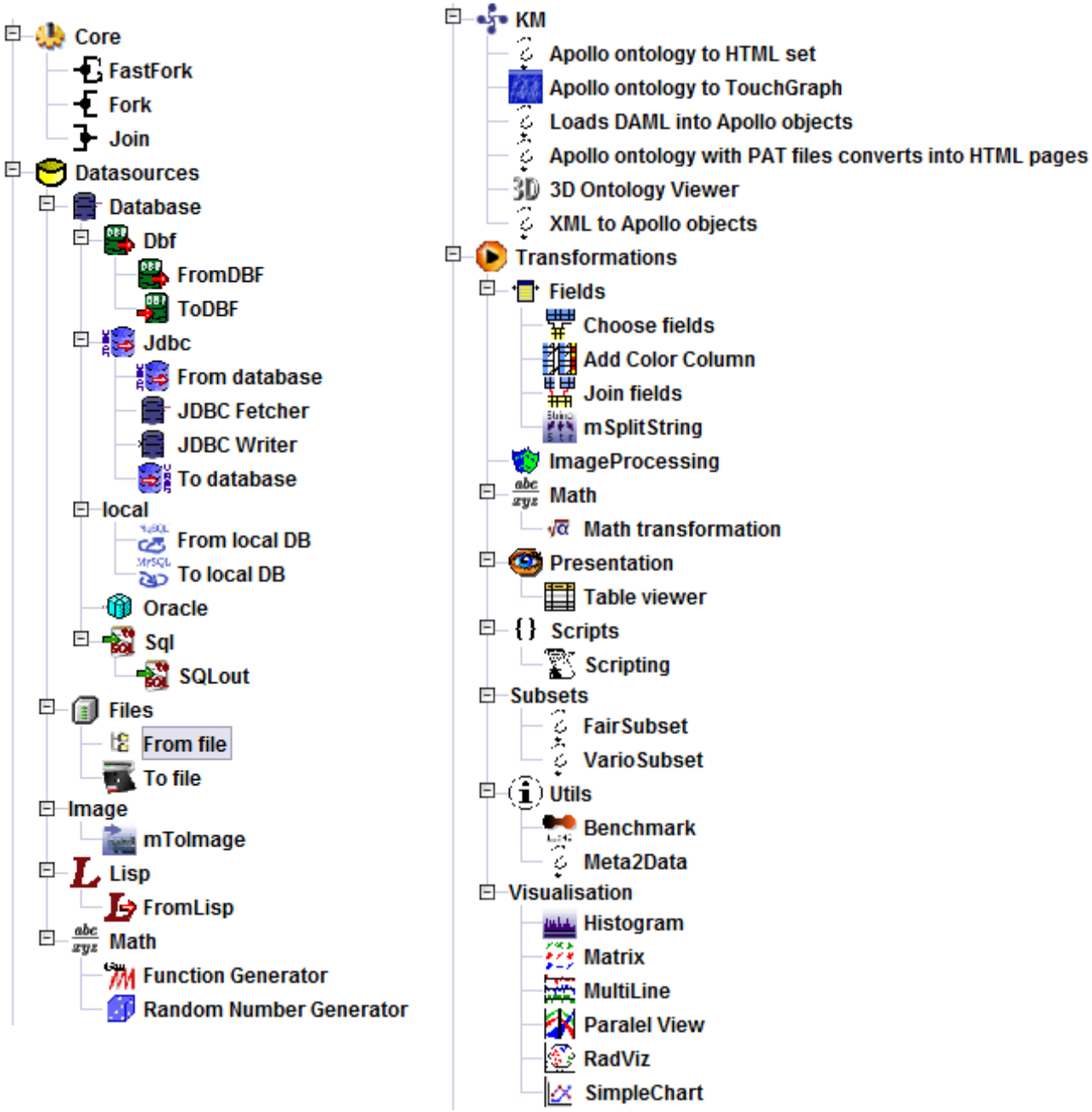
Ke spuštění Sumatry je zapotřebí mít Java Runtime 1.4 nebo vyšší. Vývojáři používají Sun JRE. Výše zmíněný archiv stačí stáhnout v sekci Download, rozbalit a spustit runx.sh (na Unix / Linux) nebo run.bat (na Windows). Je nutné ještě zmínit, že je to jen pre-verze, a ne všechny moduly jsou stabilní. V balíčku je zde také ukázkový projekt (sample1.sprj), na kterém si můžete vyzkoušet SumatraTT.



Obrázek 9.19: Pracovní plocha Sumatra

Kompletní strom modulů

Zde pouze pro představu uvádím kompletní seznam modulů, které systém Sumatra TT umožňuje použít. Jak vidíme, jedná se opravdu především o moduly pro načítání, transformování a zobrazování dat, jak již bylo v úvodu zmíněno



Obrázek 9.20. Strom modulů Sumatra

Shrnutí pojmů 9.

SW pro Data Mining.

SPSS Clementine. WEKA. SUMATRA. Statistica Data Miner. Oracle Data Mining. Microsoft SQL Server 2008. Enterprise Miner.

Otázky 9.

1. Pokud se rozhodnete vybudovat vlastní datový sklad a v něm mj. používat i metody pro Data Mining, které systémy máte možnost použít a který byste volil? Proč?
2. Co všechno bude hrát roli při rozhodování o použitém systému?

10. ZÁVĚR

Čas ke studiu: 1/2 hodiny

Cíl Přečtením této kapitoly se seznámíte s tím, že

- nejen z numerických dat je možno dolovat
- jiné datové typy se převádějí na numerická data a pak se pro ně používají již známé metody dolování

Výklad

Jiné metody pro data numerická

Zvláště pro některé specifické úlohy se vyskytuje v literatuře řada dalších metod získávání znalostí.

Mnoho snahy se věnuje zlepšování algoritmů, především v souvislosti s ohromnými rozsahy analyzovaných dat. Častým řešením u rozsáhlých dat je vzorkování. Místo celých dat se analyzují jejich části = vzorky, vybírané náhodně nebo systematicky v rozsahu, který je možno počítat bez využití vnějších pamětí pro omezení přenosů dat mezi vnitřní pamětí a diskem. Výsledek může být považován za odhad celkového výsledku, pokud dostačuje nižší spolehlivost výsledku. Jindy je po vzorcích zpracován celý soubor, z každého vzorku jsou vybráni kandidáti nebo reprezentanti pro další analýzu a s nimi je znovu proveden výpočet.

Data textová

Užitečné znalosti jsou samozřejmě rozptýleny nejen ve strukturovaných databázích, ale v mnohých dalších zdrojích, především v textech – článcích, knihách, zprávách, textových databázích. Vyhledávat informace a získat z nich zobecněné znalosti je úloha náročnější, než zpracovat strukturovaná data. Jednou z možností získávat znalosti z textů je popsat množinu textových dokumentů vhodnými numerickými parametry a na ně použít metody výše popsané.

Uveďme si jako příklad metodu pomocí strukturovaných dotazů v podobě “topiku”. Princip spočívá v porovnání textu s popisem hledané informace a vypočtení pravděpodobnosti, s jakou je obsah dokumentu k informaci relevantní. Hledaná informace se zadává pomocí klíčových slov, jejich struktury, váhy a operátorů. Struktura ve formě vyhledávacího stromu (topiku), váha je přiřazena klíčovým slovům a operátory jsou nejen klasické logické AND a OR, ale i operátor Accrue (= čím více, tím lépe), řešící klasický rozpor mezi přesností a úplností vyhledávání jen pomocí AND/OR. Vstupem metody jsou analyzované dokumenty a seznam „profilů“ pomocí jednoduchého seznamu klíčových slov nebo strukturovaným topikem. Výsledkem je strukturovaná matice, obsahující v řádcích nalezené relevantní dokumenty, ve sloupcích relevanci dokumentu odpovídající jednotlivým profilům. Nabývají hodnot 0 – 100. Na tyto matici je možno použít shlukování, hledání asociací, klasifikace apod.

Data grafická, zvuková, nestrukturovaná

Obdobně i pro další data multimediální jsou vyvíjeny metody, jak je převést na datovou numerickou matici s pevným počtem atributů. I zde jde o jistý druh komprese informace, účelem metod je snaha o minimalizaci její ztráty při strukturovaném popisu informace.

Jako příklad uveďme typ dat, který se podle významu informace nazývá sekvencí, signálem, časovou, dimenzionální nebo jednorozměrnou řadou apod. Sekvence přísluší zkoumané proměnné jako jeden z jejích nestrukturovaných atributů. Jde o posloupnost dvojic {t,X}, kde t je nezávislou dimenzí a X={x1, x2, …, xn} je množinou vlastností příslušné entity v konkrétním bodě nezávislé dimenze. Jako příklad uveďme naměřenou sekvenci EEG pacienta. Chceme-li taková data analyzovat, opět jednou z možností je charakterizovat sekvenci pevným vektorem charakteristik (min, max, průměr, šikmost, …), přičemž zřejmě množina takových charakteristik bude pro každý typ sekvence různá. Některé typy sekvencí (např. právě EEG, EKG, …) je možno považovat za "periodické". Pak lze rozdělit celou sekvenci na podsekvence (též grafoelementy), charakterizovat každou z nich samostatnou n-ticí atributů a analyzovat je jako množinu entit: shlukováním vyhledávat odchylky apod.

Výhled do blízké budoucnosti

Mimo klasické databázové zdroje – cíleně sebraná data nebo relační, transakční a objektově-relační databáze - se postupně stále častěji objevují snahy o rozšíření této základny. Již jsme zmínili data databázová multimediální všech typů. V blízké budoucnosti se bude zřejmě stále častěji obracet pozornost na využití ohromného potenciálu informací na Internetu ve všech jeho podobách.

Shrnutí pojmů 10.

Dolování v datech nenumerických. Převod dat nenumerických na numerickou reprezentaci.

Data textová, grafická, zvuková.

LITERATURA

[1] ANDĚL, Jiří. *Matematická statistika*. Praha: SNTL, 1985. ISBN 04-003-85.

[2] BERKA, Petr. *Dobývání znalostí z databází*. Praha: Academia, 2003. ISBN 80-200-1062-9.

[3] BERKA, Petr. *Tvorba znalostních systémů*. VŠE v Praze, 1993. ISBN 78-8162-841-6.

[4] BLAHUŠ, Petr. *Faktorová analýza a její zobecnění*. Praha: SNTL, 1985. ISBN 0402885.

[5] FORGY, E., *Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications*. Biometrics 21:768. 1965.

[6] GÓRECKI, Jan. *Krasobruslaři*. Ostrava, 2011. Semestrální práce. VŠB - TUO. Vedoucí práce Jana Šarmanová

[7] HÁJEK, Petr, Tomáš HAVRÁNEK a Metoděj K. CHYTIL. *Metoda GUHA*: *Automatická tvorba hypotéz*. Praha: Academia, 1983. ISBN 80-200-1062-9.

[8] HAN, Jiawei a Micheline KAMBER. *Data Mining: Concepts and Techniques*. San Francisco: Morgan Kaufmann Publishers, 2006. ISBN 13-978-1-5586-901-3.

[9] KOTÁSEK, Petr a Jaroslav ZENDULKA. *AprioriItemset - A New Algorithm for Discovering Frequent Itemsets*, *Proceedings of the MOSIS*, 1999

[10] LACKO, Ladislav. *Datové sklady, analýza OLAP a dolování dat.* Praha: Computer Press, 2003. ISBN 80-7226-969-0.

[11] LUKASOVÁ, Alena a Jana ŠARMANOVÁ. *Metody shlukové analýzy*. Praha: SNTL, 1985. ISBN 04-014-85

[12] PARR RUD, Olivia. *Data Mining*. Brno: Computer Press, 2001. ISBN 80-7226-577-6.

[13] PYLE, Dorian. *Data Preparation for Data Mining*. San Francisco: Morgan Kaufmann Publishers, Inc., 1999. ISBN 1-55860-529-0.

[14] REISENAUER, Roman. *Metody matematické statistiky*. Praha: SNTL, 1965.

[15] ROJÍČEK, R. *Systém automatizovaného získávání znalostí od experta.* Ostrava, 1999. 68 s. Diplomová práce. VŠB-TU Ostrava, FEI.

[16] TROSZOK, Aleš. *Informační a komunikační technologie v procesu pedagogického hodnocení*. Ostrava, 2011. Disertační práce. Ostravská univerzita. Vedoucí práce Josef Malach.

[17] Weka (machine learning). In [online]. [s.l.] : [s.n.], 2. 2.2011 [cit. 2011-03-23+. Dostupné z WWW: <http://en.wikipedia.org/wiki/Weka_(machine_learning)>.

[18] ZENDULKA, Jaroslav, Vladimír BARTÍK, Roman LUKÁŠ a Ivana RUDOLFOVÁ. *Získávání znalostí z databází*: *Studijní opora* [online]. Brno: FIT VUT, 2009.

[19] ŽIŽKA, Jan. *Velmi stručný úvod do použití systému WEKA pro Data Mining*. In [online]. [s. l.] : [s. n.], 10. 3. 2009 [cit. 2011-03-23+. Dostupné z WWW:

<https://akela.mendelu.cz/~zizka/Machine.../WEKA_Data-mining.pdf>.