VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

ZÍSKÁVÁNÍ ZNALOSTÍ Z DAT - SHLUKOVACÍ ALGORITMY

BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE RADIM KAPAVÍK
AUTHOR

BRNO 2008

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

ZÍSKÁVÁNÍ ZNALOSTÍ Z DAT – SHLUKOVACÍ ALGORITMY

KNOWLEDGE DISCOVERY FROM DATA - CLUSTERING ALGORITHMS

BAKALÁŘSKÁ PRÁCE
BACHELOR´S THESIS

AUTOR PRÁCE RADIM KAPAVÍK
AUTHOR

VEDOUCÍ PRÁCE ING. VLADIMÍR BARTÍK, PH.D.
SUPERVISOR

BRNO 2008

Zadání bakalářské práce/7046/2007/xkapav01

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav informačních systémů Akademický rok 2007/2008

Zadání bakalářské práce

Řešitel: **Kapavík Radim**
Obor: Informační technologie
Téma: **Získávání znalostí z dat - shlukovací algoritmy**
Kategorie: Databáze

Pokyny:
1. Seznamte se s problematikou získávání znalostí z dat, zaměřte se na shlukování.
2. Po dohodě s vedoucím zvolte algoritmus a navrhněte aplikaci, která by vhodným způsobem prezentovala funkčnost a výsledky zvoleného algoritmu.
3. Navrženou aplikaci implementujte a ověřte její funkčnost na zvoleném vzorku dat.
4. Zhodnoťte dosažené výsledky a další možná pokračování tohoto projektu.

Literatura:
- Zendulka, J. a kol.: Získávání znalostí z databází - studiní opora. FIT VUT Brno, 2006.
- Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, 2001.

Při obhajobě semestrální části projektu je požadováno:
- Body 1-2

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Bartík Vladimír, Ing., Ph.D.**, UIFS FIT VUT
Datum zadání: 1. listopadu 2007
Datum odevzdání: 14. května 2008

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
612 66 Brno, Božetěchova 2

L.S.

doc. Ing. Jaroslav Zendulka, CSc.
vedoucí ústavu

LICENČNÍ SMLOUVA
POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami

1. Pan

Jméno a příjmení: **Radim Kapavík**
Id studenta: 78947
Bytem: Hranická 12, 751 24 Přerov
Narozen: 19. 11. 1985, Přerov
(dále jen "autor")

a

2. Vysoké učení technické v Brně

Fakulta informačních technologií
se sídlem Božetěchova 2/1, 612 66 Brno, IČO 00216305
jejímž jménem jedná na základě písemného pověření děkanem fakulty:

....
(dále jen "nabyvatel")

Článek 1
Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):
   bakalářská práce

Název VŠKP: Získávání znalostí z dat - shlukovací algoritmy
Vedoucí/školitel VŠKP: Bartík Vladimír, Ing., Ph.D.
Ústav: Ústav informačních systémů
Datum obhajoby VŠKP: ....

VŠKP odevzdal autor nabyvateli v:
tištěné formě počet exemplářů: 1
elektronické formě počet exemplářů: 2 (1 ve skladu dokumentů, 1 na CD)

2. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
3. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
4. Autor potvrzuje, že listinná a elektronická verze díla je identická.

Článek 2
Udělení licenčního oprávnění

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti:
   - ☐ ihned po uzavření této smlouvy
   - ☐ 1 rok po uzavření této smlouvy
   - ☐ 3 roky po uzavření této smlouvy
   - ☐ 5 let po uzavření této smlouvy
   - ☐ 10 let po uzavření této smlouvy
   (z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

Článek 3
Závěrečná ustanovení

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísni a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: ....

....	...............Kapoun...............
Nabyvatel	Autor

Abstrakt

Tato práce se zabývá shlukovou analýzou se zaměřením na problémy stanovení nezbytných parametrů shlukování. Její převážná část je věnována popisu implementace metody DENCLUE, založené na hustotě, a návrhu způsobu automatického nastavování jejího klíčového parametru označovaného jako σ.

Klíčová slova

Shluková analýza, shlukování, shlukovací algoritmy, DENCLUE

Abstract

This work deals with the theme of cluster analysis, focusing on problems of determining necessary parameters of these methods. Most of the work is dedicated to describing implementation of DENCLUE method based on density and proposing appropriate way to set up it´s key parameter, known as σ, automatically.

Keywords

Cluster analysis, clustering, clusterging algorithms, DENCLUE

Citace

Kapavík Radim: Získávání znalostí z dat – shlukovací algoritmy. Brno, 2008, bakalářská práce, FIT VUT v Brně.

Získávání znalostí z dat – shlukovací algoritmy

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Vladimíra Bartíka, Ph.D.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……………………
Radim Kapavík
7.5. 2008

Poděkování

Děkuji Ing. Vladimíru Bartíkovi, Ph.D. za odbornou pomoc při vytváření této práce.

Obsah

Úvod

První kapitola stručně nastiňuje co je to získávání znalostí z dat a k čemu se využívá.

Ve druhé kapitole pojednávám o shlukové analýze a jejím využití a nastiňuji několik nejobvyklejších metod shlukové analýzy, resp. tříd metod – metody založené na rozdělování, metody hierarchické a metody založené na hustotě, speciálně metodu DENCLUE.

Ve třetí kapitole pak popisuji implementaci metody DENCLUE. Po definování cíle implementace následuje teoretický rámec, kterým je popis různých možných typů atributů a práce s nimi a metody výpočtu vzdálenosti objektů, načež popisuji konkrétní implementaci, od přípravy dat, přes výpočet vzdálenosti objektů a funkce hustoty nad nimi, až po samotné shlukování a výstup programu. Poslední částí třetí kapitoly je diskuze nad možnostmi nastavení parametrů metody DENCLUE a vysvětlení návrhu řešení nastavení parametru σ, použitého při implementaci.

Čtvrtá kapitola obsahuje ukázky testovacích dat, na nichž byla ověřena funkčnost programu.

Pátá kapitola ve formě závěru shrnuje výsledky a navrhuje některé úpravy, které by bylo vhodné provést při použití programu v praxi.

1 Získávání znalostí z dat

1.1 Co je získávání znalostí z dat

Získávání znalostí z databází je proces, který umožňuje ve velkém množství dat nalézt zajímavé informace, které dosud nebyly známy*. V současné době, kdy v informační společnosti často už není problém v nedostatku dat, ale naopak v jejich obrovském množství, se stává čím dál důležitější oblastí informatiky, neboť umožňuje data efektivně využít.

Získávání znalostí je proces, který – i když je na statistice do značné míry založen – umožňuje více než statistické metody; získávání znalostí umožňuje v datech (zpravidla velkém množství dat) automaticky nalézt potenciálně užitečné vzory a modely. Tím dochází k nalezení netriviálních, dosud neznámých skrytých znalostí, k jejichž uchovávání není databáze určena (nelze je tedy například získat sebekomplexnějším SQL dotazem nad databází).

Fázi samotného dolování dat, která tvoří jádro procesu, a jejímž výsledkem je právě nalezení dosud neznámých znalostí, zpravidla předchází několik přípravných fází (předzpracování dat). Data použitá při získávání znalostí jsou zpravidla uložena v datových skladech, které obsahují velké množství dat, často shromážděných za delší období a z několika různých zdrojů. První fází je tak čistění a integrace dat. Jejím úkolem je jednak vypořádat se se šumem a jednak zajistit konzistenci dat pocházejících z více zdrojů a různých dob.

Druhou fází procesu je výběr dat. Jeho úkolem je identifikovat data, která jsou pro zadanou úlohu relevantní. Jde v praxi o výběr sloupců databáze či dimenzí datové kostky.

Třetí fází je transformace dat – jejím cílem je zmenšit objem dat tím, že se vhodně seskupí (např. agregace, sumarizace, shlukování).

Následuje samotné dolování dat. Je vhodné upozornit, že dolování dat poskytne podklady pro rozhodnutí, ale jejich interpretace je záležitostí myslícího člověka. Jeho úkolem je identifikovat pro něj užitečné výsledky a provést rozhodnutí, opírající se o tyto výsledky.

1.2 Využití získávání znalostí z dat

Nejširší uplatnění techniky získávání znalostí z dat nachází v oblasti marketingu. Typickými příklady jsou analýza nákupního košíku a analýza trhu. Analýza nákupního košíku je proces hledání vzorů v provedených nákupech za účelem předvídání chování zákazníka. Typickým výsledkem je zjištění, že zákazníci kupující výrobek X si často kupují i výrobek Y, a nebo (s rozšířením o časové údaje) zjištění závislosti poptávky po určitém zboží na datu či čase. Výsledkem analýzy trhu může být zjištění profilu zákazníků, tedy kteří zákazníci jak nakupují (viz kapitola 2.2).

Dalšími oblastmi, kde se získávání znalostí uplatňuje, jsou úlohy predikce vývoje rizika či cen, získávání znalostí z textu se dá využít např. pro filtrování nevyžádané pošty, vyhledávání neobvyklých vzorů je pak využitelné v oblasti zabezpečení (viz kapitola 2.2). Není pochyb o tom, že s těmito metodami nějakým způsobem pracují i tajné služby.

* Při vytváření první kapitoly jsem čerpal především z [2]

2 Shluková analýza

2.1 Definice pojmu

Shlukování je proces rozdělování objektů do tříd (shluků) na základě své podobnosti. Podobnost je definována na základě podobnosti hodnot jednotlivých atributů objektů. Podobnost atributů se zpravidla určuje na základě vzdálenostní funkce [2]. V tomto grafickém vyjádření (které je asi nejnázornější) pak atributy představují hodnoty v n-rozměrném prostoru (kde n je rovno počtu atributů).

Shlukování je přirozeným způsobem práce s daty, které každý člověk používá. Všechny předměty ve světě kolem nás si automaticky rozdělujeme do tříd na základě své podobnosti. Některé jsou jasněji definovatelné (např. rozdělení lidí na muže a ženy), jiné hůře (např. rozdělení filmů podle žánru); ve skutečnosti neexistuje pro daný soubor objektů jednoznačně „správné" rozdělení do shluků.

Stejně důležité jako shluky podobných objektů jsou i tzv. odlehlé hodnoty, tedy objekty, které se nedaří zařadit do žádného shluku, neboť jejich atributy se výrazně liší od všech ostatních.

Nejnázornějším je shlukování v prostoru, jak už jsem zmínil, například několik bodů v rovině dokáže každý automaticky rozdělit do oddělených skupin bodů, které jsou blízko u sebe. Implementovat tento postup automaticky za pomoci výpočetní techniky však není triviální.

2.2 Využití shlukové analýzy

Jak jsem uvedl v kapitole 1.1, shlukování může být součástí třetí fáze procesu získávání znalostí z dat (transformace dat). Cílem shlukování je – stejně jako u ostatních výše zmíněných metod – zmenšit počet dat vstupující do dalšího procesu získávání znalostí z dat, konkrétně tím, že podobná data jsou označena za shluk a nadále jsou reprezentována jednou hodnotou.

Shluková analýza se však uplatňuje i samostatně. V rámci analýzy trhu například v hledání skupin (shluků) zákazníků, kteří se chovají (nakupují) podobně. Tyto výsledky lze pak velmi dobře využít například pro cílenou reklamní kampaň. Druhým způsobem využití shlukové analýzy je naopak hledání neobvyklých, podezřelých objektů, tedy takových, které se významně liší od většiny ostatních – např. detekce neobvyklého pohybu osoby ve sledovaném prostoru může spustit výstražný systém[*].

2.3 Metody shlukování

Shlukování se z hlediska strojového učení řadí k metodám učení bez učitele[†]. Existuje celá řada metod shlukování, které by měly všechny splňovat několik kritérií:

---

[*] V oblasti zabezpečení se s takovýmito metodami pracuje velice opatrně. Je zde nutno jednak používat zcela spolehlivé a vyzkoušené metody, a jednak pamatovat na mimořádné situace – bylo by jistě velmi nešťastné, kdyby výstražný systém, který by kupříkladu zablokoval některý průchod či funkci určitého zařízení, svým neobvyklým chováním spustil pracovník, řešící mimořádnou situaci.

[†] Popis metod shlukování v této práci vychází především z [1] a [2]

Škálovatelnost – shluková analýza se většinou provádí na velkém množství dat, proto by algoritmus měl být schopen vypořádat se s velkým množstvím dat stejně jako s malým

Schopnost pracovat s různými typy dat – v praxi jsou parametry objektů nejen čísly, ale někdy též binární hodnotou, ordinální hodnotou apod.; algoritmus by měl být schopen zpracovat data různého typu (a to i dohromady)

Schopnost najít shluky libovolného tvaru – algoritmus by neměl preferovat určitý tvar shluku, ale najít vždy takové, jaké nejlépe odpovídají datům

Schopnost vypořádat se s šumem a odlehlými hodnotami – chybějící nebo neznámé hodnoty by neměly zkreslit výsledky

Schopnost zpracovávat data o velkém počtu atributů – data o dvou až třech atributech (dimenzích) je možné analyzovat při grafickém znázornění pouhým okem; shlukovací algoritmy by měly být schopny pracovat stejně dobře s daty o více atributech

Možnost provést shlukování při zadání co nejméně dodatečných informací – největším problémem všech shlukovačích algoritmů je nastavení různých parametrů, které ke svému průběhu vyžadují od uživatele, a které mohou výrazně ovlivnit výsledek shlukování; na tento faktor se budu v celé práci speciálně zaměřovat.

2.3.1 Metody založené na rozdělování

Metody založené na rozdělování jsou nejjednodušší shlukovací metody (metoda K-Means byla představena roku 1967[*]), pracují tak, že rozdělují objekty do *k* tříd. Principem je najít *k* center shluků tak, aby suma čtverců vzdáleností jednotlivých bodů, které jsou přiřazeny vždy k nejbližšímu centru shluku, byla co nejnižší.

V prvním kroku jsou centra shluků zvolena libovolným způsobem (většinou náhodně). Následně jsou všechny body přiřazeny k nejbližšímu centru a je nově vypočítán střed takto vzniklých shluků. U metody K-Means je to fiktivní centrální objekt, jehož atributy jsou vypočítány jako střední hodnota atributů objektů příslušejících danému shluku, u metody K-Medoids je to skutečný objekt, který je vypočítanému fiktivnímu středu nejblíže (tím se dosahuje snížení vlivu odlehlých hodnot na výsledek shlukování). Následuje opětovné přiřazení objektů k novým středům shluků, kterýžto proces se opakuje, dokud se při přepočítávání poloha středů mění.

Existují i varianty tohoto algoritmu, které neřadí objekt vždy jen do jednoho ze shluků, ale do více zároveň (FCM – Fuzzy C-Means).

Nevýhodami těchto metod je to, že nenaleznou vždy řešení optimální, nýbrž často jen suboptimální, že nalézají pouze shluky kulového tvaru a že výsledek shlukové analýzy závisí na volbě počátečního umístění center shluků.

Co se týče zadávaných parametrů, metoda vyžaduje zadání počtu shluků, což je její největší slabina. Předpokládá totiž, že už při spuštění známe (aspoň částečně) výsledek shlukování.

[*] MacQueen, J. B. Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability. Berkeley, University of California Press, 1967, s. 281-297

2.3.2 Hierarchické metody

Základní hierarchická metoda byla představena ve stejném roce jako metoda K-Means*, představuje však výrazně jiný přístup ke shlukování. Hierarchické metody provádějí hierarchický rozklad množiny objektů a vytvářejí tak strom shluků, kdy na nejnižší úrovni je každý objekt zvláštním shlukem a na nejvyšší *n*-té úrovni (kde *n* je počet objektů) tvoří všechny objekty jeden shluk. V každé *i*-té úrovni mezi těmito dvěma extrémy je pak počet shluků o jeden nižší než v úrovni *i*-1.

Existují metody shlukující (zdola nahoru) a rozdělující (shora dolů); v praxi se používají většinou shlukující. Ty začínají tím, že každý objekt považují za shluk (nejnižší úroveň) a vypočítají vzdálenosti mezi každou dvojicí shluků. V každém následujícím kroku pak spojí dva nejbližší shluky a nahradí je novým, pro nějž přepočítají vzdálenost ke každému z ostatních shluků. Končí dosažením stavu, kdy existuje jediný shluk (tj. nejvyšší úrovně), případně dosažením počtu shluků zadaného uživatelem.

Výhodou této metody oproti předchozí je, že nepožaduje předem zadat počet shluků, ale umožňuje uživateli ze vzniklé hierarchie shluků vybrat úroveň, která mu vyhovuje. Na druhou stranu díky postupnému tvoření shluků není možno jednou spojené shluky v dalších krocích rozdělit ani přesunout objekty mezi shluky (i když by to mnohdy vedlo k lepšímu výsledku). Výsledky, které tato metoda produkuje, proto nemusí být nutně pro daný počet shluků optimální.

Co se týče zadávání parametrů, metoda nevyžaduje vložení žádných parametrů, namísto toho je třeba provést výběr vhodné úrovně z hierarchie výsledných shluků. To je pro uživatele jistě příjemnější způsob jak zvolit počet shluků. Stejného efektu, bez zmíněného nebezpečí nepřesnosti výsledků, lze dosáhnout několikanásobným použitím jiné metody (např. K-Means) s postupně měněným parametrem udávajícím požadovaný počet shluků.

2.3.3 Metody založené na hustotě

Metody založené na hustotě považují objekty za body v *n*-rozměrném, kde *n* je počet atributů objektů, a umístění jednotlivých objektů v tomto prostoru pak odpovídá hodnotám atributů. Za shluky jsou považovány oblasti v tomto prostoru s velkou hustotou objektů oddělené oblastmi s malou hustotou objektů (jednotlivé objekty mezi shluky jsou považovány za šum nebo odlehlé hodnoty).

2.3.3.1 Metoda DBSCAN

Metoda DBSCAN (Density-Based Spatial Clustering of Applications with Noise), navržená v roce 1996, je asi nejznámějším zástupcem metod založených na hustotě. Pracuje s pojmem okolí bodu o poloměru ε, tedy tzv. ε-okolí. Jestliže ε-okolí bodu (objektu) obsahuje stanovený minimální počet objektů, je objekt označen za centrum shluku. Následně do shluku přidá všechny objekty, které se nacházejí v ε-okolí centra shluku, což vede zpravidla ke spojování shluků.

Nevýhodou této metody je nutnost nastavit parametry (ε a minimální počet objektů potřebný k tomu aby byl objekt prohlášen za centrum shluku). Tyto parametry ovlivňují počet shluků, které budou nalezeny, toto nastavení je však ještě méně průhledné než u metod, kde se počet shluků nastavuje přímo.

---

* Johnson, S.C. Hierarchical Clustering Schemes. Psychometrika, 2, 1967, s. 241-254

2.3.3.2 Metoda DENCLUE

Metoda DENCLUE (Density-based Clustering) provádí shlukování na základě funkce hustoty. Vliv každého objektu na své okolí je modelován matematickou funkcí, a funkce hustoty, definovaná nad celým prostorem dat, je součet všech funkcí vlivu (hodnota funkce hustoty v každém bodě je dána součtem funkcí vlivu jednotlivých objektů v daném bodě). Shluky jsou pak místa, kde tato funkce hustoty dosahuje lokálního maxima.

Funkce vlivu je jakákoliv funkce odvozená od vzdálenosti objektů, v nejjednodušším případě je definovaná tak, že její hodnota je 1 ve vzdálenosti menší než σ a 0 ve vzdálenosti větší než σ. Podrobnější popis viz kapitola 3.3.3.

Problémem metody je stanovení parametru σ. Možné řešení nastiňuje kapitola 3.4.3.

3 Implementace metody DENCLUE

3.1 Specifikace cíle

Cílem této bakalářské práce je navrhnout a vytvořit program schopný pomocí metody DENCLUE provést shlukovou analýzu zadaných dat (objektů). Program by měl být schopen zpracovávat data o mnoha rozměrech (objekty o mnoha atributech) a data různého typu – tedy nejen reálná čísla. Vstupem je soubor v textovém formátu a výstupem je také jednoduchý textový formát (grafické znázornění shluků o více rozměrech je prakticky nemožné). Cílem programu je demonstrovat funkčnost metody DENCLUE a především navrženého řešení problému stanovení parametrů této metody.

3.2 Teoretická východiska

3.2.1 Typy atributů

Atributy objektů mohou nabývat různých typů.

Základním je **intervalová** proměnná, což je spojité měřítko, tedy reálné číslo. Pro intervalové atributy se odlišnost objektů vyjadřuje jako jejich vzdálenost. Obecně lze použít jakoukoliv vzdálenostní funkci, nejčastěji se používá klasická euklidovská vzdálenost:

$$d_{ij} = \sqrt{\left|x_{i1} - x_{j1}\right|^2 + \left|x_{i2} - x_{j2}\right|^2 + \ldots + \left|x_{in} - x_{jn}\right|^2}\ ,$$

kde $i = (x_1, x_2, \ldots, x_n)$ a j = $(x_1, x_2, \ldots, x_n)$ jsou dva n-dimenzionální objekty.

Dalším typem proměnné je proměnná **binární**, která může nabývat jen dvou hodnot (označovaných většinou 0 a 1). S těmito hodnotami nelze pracovat stejně jako s intervalovými proměnnými. Binární proměnné lze rozdělit na dva typy – na binární proměnné symetrické a binární proměnné asymetrické.

U symetrických binárních proměnných mají oba stavy stejnou pravděpodobnost a tudíž mají stejnou váhu. Vzdálenost objektů s binárními atributy lze vypočítat jako:

$$d_{ij} = \frac{r}{r + s},$$

kde r je počet proměnných, které mají pro objekt i jinou hodnotu než pro objekt j a s je počet proměnných, které mají pro oba objekty hodnotu stejnou.

Asymetrické binární proměnné jsou takové, jejichž jedna hodnota je významnější než druhá (má menší pravděpodobnost výskytu) – například vyhodnocování atributů představujících koníčky jednotlivých lidí, kdy shoda v pozitivní hodnotě je významnější než v negativní. Vzdálenost se pak určuje jako:

$$d_{ij} = \frac{r}{r+t},$$

kde r je počet proměnných, které mají pro objekt i jinou hodnotu než pro objekt j a t je počet proměnných, které mají pro oba objekty hodnotu 1 (pozitivní shoda) – negativní shody se neberou v potaz.

Zobecněním binárních proměnných jsou proměnné **nominální**, které mohou nabývat více než dvou hodnot. Zpravidla nejsou číselné, ale vyjádřené slovně. Vzdálenost proměnných s atributy tohoto typu lze spočítat jako:

$$d_{ij} = \frac{p-m}{p},$$

kde p je celkový počet proměnných a m je počet proměnných které mají stejnou hodnotu pro oba objekty.

Dále existují ještě proměnné **ordinální**, které jsou podobné nominálním s tím rozdílem, že hodnoty, kterých mohou nabývat, jsou seřazeny v určitém pořadí – jejich vzdálenost se počítá tak, že jejich hodnoty nahradíme číselnými hodnotami tak aby tyto odpovídaly jejich seřazení a použijeme stejný postup jako u intervalových proměnných. Případně můžeme za zvláštní typ proměnných považovat i proměnné **poměrové**, které jsou hodnotami vyjádřenými na nelineární stupnici (často exponenciální). S těmi se pracuje buď stejně jako s lineárními (poté co se transformují do lineární stupnice, případně i bez toho), nebo jako s ordinálními.

3.2.2 Výpočet vzdálenosti objektů s atributy různých typů

V praxi je běžné, že objekty, nad nimiž je prováděno shlukování, mají atributy více různých typů. Je tudíž třeba při výpočtu vzdálenosti objektů kombinovat výše uvedené postupy pro jednotlivé typy proměnných tak, aby nedocházelo ke zkreslení výsledků. Vzdálenost dvou objektů s atributy různých typů lze vypočítat podle tohoto vztahu:

$$d_{ij} = \frac{\sum_{f=1}^{n} \sigma_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^{n} \sigma_{ij}^{(f)}},$$

kde $\sigma_{ij}^{(f)} = 0$ jestliže příslušný atribut u jednoho z objektů chybí a nebo jestliže jde o atribut typu asymetrická binární proměnná a obě hodnoty jsou nulové (negativní shoda) a $d_{ij}^{(f)}$ je vzdálenost objektů vypočítaná podle f-tého atributu.

Hodnoty $d_{ij}^{(f)}$ je třeba normalizovat tak, aby při shlukování měly všechny atributy stejnou váhu a nedošlo k deformaci prostoru dat způsobené tím, že rozsah hodnot různých proměnných je různý. Většinou se hodnoty transformují do intervalu <0,1> (hodnoty binárních proměnných tuto podmínku splňují bez transformace) tímto způsobem:

$$d_{ij}^{(f)} = \frac{\left| x_{if} - x_{jf} \right|}{\max_h x_{hf} - \min_h x_{hf}},$$

kde h jde přes všechny hodnoty proměnné f pro jednotlivé objekty.

3.3 Popis implementace

3.3.1 Příprava dat

3.3.1.1 Vstup

Vstupní data programu jsou vkládána v jednoduchém textovém souboru určeného formátu, který umožňuje uživateli zadat libovolný počet objektů s libovolným počtem parametrů. Každý objekt má název a seznam atributů. V úvodní deklaraci je třeba specifikovat názvy a typy atributů. Podrobný popis formátu vstupního souboru viz příloha č. 1.

3.3.1.2 Normalizace dat

První operací, kterou je nutno provést s načtenými daty, je normalizace hodnot atributů, aby bylo možno účinně pracovat s atributy různých typů a různých rozsahů.

Nejdříve je nutno atributy typu nominální proměnné převést na číselné hodnoty, a to tím způsobem, že každá nová (textová) hodnota vyskytující se ve vstupním souboru dat je mapována na celé číslo o 1 vyšší než nejvyšší už použité. Konkrétní hodnoty nejsou důležité, neboť nominální proměnné nejsou seřazeny, nelze rozhodnout, která je větší a která menší, rozlišuje se pouze rovnost nebo nerovnost této hodnoty u dvou objektů; důležité je pouze zachovat mezi nimi stejné rozestupy, v tomto případě konkrétně 1.

Po načtení všech vstupních dat a převedení nominálních proměnných na čísla je možno provést samotnou normalizaci podle vztahu uvedeného v kap. 3.2.2. Je třeba zjistit interval, do něhož spadají všechny hodnoty jednotlivých atributů (tedy nejvyšší a nejnižší hodnotu). Tento interval se pak považuje za rozsah <0;1> a všechny reálné hodnoty jsou do něj mapovány na příslušná místa. Vznikne tak normalizovaný prostor dat jakožto n-rozměrná krychle o straně 1.

V dalším průběhu shlukování už se používají výhradně tyto normalizované hodnoty, až na výstup jsou vytištěny opět hodnoty původní.

3.3.2 Výpočet vzdálenosti objektů

Pro další průběh shlukování je třeba znát vzdálenosti jednotlivých objektů. Toto je výpočetně a časově nejnáročnější část programu – je totiž nutné spočítat vzdálenost každého objektu ke každému dalšímu objektu. Složitost tohoto kroku je proto kvadratická a závisí na počtu objektů.

Samotná operace spočtení vzdálenosti dvou objektů je poměrně snadná. Jelikož do ní vstupují už normalizované hodnoty a používám Manhattanovskou vzdálenost, je možno prostě sečíst vzdálenosti objektů v jednotlivých rozměrech – to znamená sečíst absolutní hodnoty rozdílů hodnot jednotlivých atributů daných dvou objektů (s výjimkou nominálních proměnných, kde vzdálenost nabývá pouze hodnoty 0 pokud jsou obě hodnoty daného atributu stejné a nebo hodnoty 1 pokud jsou rozdílené – nelze je tedy odečítat – a asymetrických binárních proměnných, kde je třeba ignorovat atributy nabývající shodné negativní hodnoty). Výsledek se pak normalizuje podělením počtem rozměrů, takže vzdálenosti mezi objekty nabývají opět hodnot z intervalu <0;1>.

3.3.3 Výpočet funkce hustoty

Použil jsem jednoduchou obdélníkovou funkci vlivu jednotlivých objektů na své okolí, což znamená, že funkce hustoty v jakémkoliv bodě prostoru je rovna počtu objektů, které se nacházejí ve vzdálenosti menší než σ. Ačkoliv je funkce hustoty nad prostorem dat teoreticky definována v každém bodě, nemá smysl počítat ji kdekoliv jinde než v bodech, kde se nacházejí jednotlivé objekty.

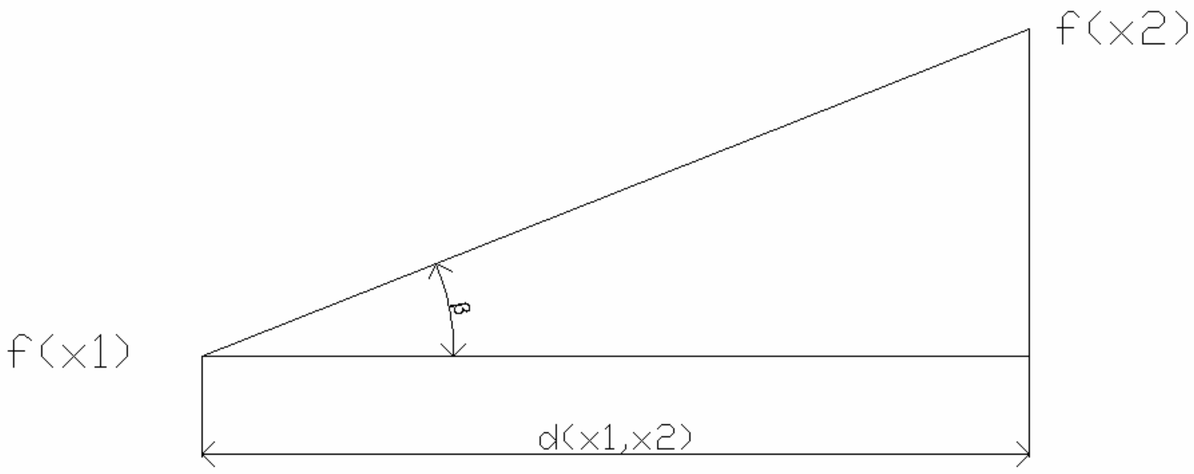
Jelikož pro každý objekt už mám spočítanou vzdálenost ke všem ostatním objektům, stačí tyto vzdálenosti seřadit od nejmenších po největší a postupně procházet, dokud není nalezen první objekt, jehož vzdálenost je větší než σ. Počet objektů se vzdáleností menší nebo rovnou σ je pak roven funkci hustoty v daném bodě neboli nad daným objektem.

3.3.4 Rozdělení objektů do shluků

Jestliže máme spočtenu hodnotu funkce hustoty pro každý objekt v prostoru dat, je možno konečně přistoupit k rozdělení objektů do shluků. Jak již bylo řečeno výše, shluky jsou lokální maxima hodnot funkce hustoty.

K nalezení lokálních maxim funkce lze s úspěchem využít „Hill Climbing“ metod (česky by se dalo přeložit nejspíše jako „metoda největšího stoupání“). Princip metody je v tom, že prohledává okolí bodu a hledá bod, v němž je hodnota funkce vyšší než v původním bodě. Pokud takový najde, považuje ho za nový výchozí bod a procedura se opakuje, dokud není nalezen bod, v jehož okolí se nenachází bod s vyšší hodnotou funkce. To je pak nalezené lokální maximum.

Poloměr okolí bodu je roven parametru σ. Ve svém programu používám variantu algoritmu, která vybere ze všech bodů v okolí ten, k němuž je stoupání nejstrmější. To brání zbytečně zdlouhavému postupu algoritmu. Strmost stoupání určuji jako směrnici spojnice hodnot funkce hustoty v obou bodech, jak ukazuje obrázek:



f(x1) je hodnota funkce hustoty v bodě x1, f(x2) je hodnota funkce hustoty v bodě x2, d(x1,x2) je vzdálenost obou bodů a úhel β, resp. jeho tangens, určuje strmost stoupání; lze ho spočíst jako poměr rozdílu f(x2) - f(x1) a d(x1,x2)

Tímto způsobem lze tedy při startu z kteréhokoliv bodu najít lokální maximum, k němuž bod přísluší. Toto lokální maximum je pak označeno za střed shluku a výchozí bod i všechny body, které se nacházejí do vzdálenosti σ od právě nalezeného nového středu shluku (zde se opět využije seřazeného seznamu vzdáleností jednotlivých objektů) je možno okamžitě přiřadit do daného shluku, neboť je jisté, že kdyby metoda nejvyššího stoupání začala od některého z nich, skončila by u toho středu shluku.

Pokud tedy projdeme všechny objekty v prostoru dat a aplikujeme na ně tuto metodu, máme jistotu, že nalezneme všechny shluky (resp. jejich středy) a zároveň každý objekt přiřadíme shluku, kterému přísluší (pokud se jedná o odlehlou hodnotu, bude tvořit samostatný shluk). Při praktické implementaci pochopitelně můžeme algoritmus optimalizovat tím, že můžeme vynechat objekty, které už byly přiřazeny v některém předchozím kroku, a navíc nemusíme vždy hledat centrum shluku, ale stačí najít jakýkoliv objekt, který už je do nějakého shluku přiřazen, neboť je jisté, že od něj už by algoritmus hledající střed shluku postupoval stejnou cestou jako při jeho přiřazování a našel by tedy tentýž střed.

3.3.5 Výstup

Zásadní informace pro uživatele je, kolik vzniklo shluků, jaké jsou jejich středy a které objekty jsou přiřazeny do kterého shluku. Proto jsem výstup rozdělil na tři části, v první program vypisuje počet nalezených shluků a použitý parametr σ (viz kap. 3.4.), ve druhé jednotlivá centra shluků (seřazená od nejvýznamnějšího po nejméně významné – odlehlé hodnoty) včetně hodnot všech svých atributů a počet objektů, které daný shluk tvoří, a ve třetí kompletní výpis veškerých objektů a všech jejich atributů roztříděných do shluků.

3.4 Nastavení parametrů

3.4.1 Parametry metody

V popisu implementace algoritmu DENCLUE se vyskytuje stěžejní parametr této metody, σ, který určuje, jak moc objekty ovlivňují své okolí, resp. do jaké vzdálenosti ho ovlivňují. Různé nastavení tohoto parametru vede k různým počtům shluků ve výsledku. Nastavení σ je navíc pro uživatele nepříliš průhledné a těžko určit, podle čeho by ho měl nastavovat – nelze dost dobře odhadnout, ke kolika shlukům jaká hodnota σ povede.

Výsledek však lze ovlivnit také parametrem, určujícím, jaké hodnoty musí minimálně dosáhnout funkce hustoty pro daný objekt, aby mohl být označen za centrum shluku (objekty, které jsou přiřazeny k lokálním maximům o hodnotě nižší než ε, jsou považovány za odlehlé hodnoty).

3.4.2 Parametr ε

Význam parametru ε není zásadní. Jeho účelem je odfiltrovat odlehlé hodnoty z výsledku. Nicméně účelem shlukování někdy může být právě nalezení zvláštních nebo podezřelých hodnot, tedy odlehlých hodnot.

Proto se domnívám, že není problém parametr ε neuvažovat, respektive považovat ho za nulový. To znamená, že ve výsledku se objeví všechny shluky, jakkoliv malé. Je pak na uživateli, aby si sám vybral ty, které jsou pro něj relevantní a neuvažoval ty, které mu přijdou už jako příliš bezvýznamné. K tomu mu pomůže seřazení výsledků od shluků, jejichž střed má nejvyšší hodnotu funkce hustoty k těm, které mají hodnotu nejnižší. Na konci seznamu se tak budou nacházet odlehlé hodnoty ve formě shluků o jednom objektu (nebo malém počtu objektů).

3.4.3 Parametr σ

Parametr σ je pro metodu DENCLUE zásadním údajem. Jeho nastavení přímo (i když ne zcela průhledným způsobem) rozhoduje o počtu shluků, do nichž budou objekty rozděleny. V případě nastavení σ=0 bude každý objekt tvořit samostatný shluk, v případě nastavení na maximální hodnotu, což je v případě normalizovaných hodnot atributů, se kterými tato implementace pracuje, hodnota σ=1, budou naopak všechny objekty tvořit jeden shluk. Cílem je najít vhodnou hodnotu σ někde mezi těmito dvěma extrémy.

Obecně nelze určit, jaká hodnota σ je správná. To vyplývá z toho, že neexistuje nic jako „správné“ rozdělení objektů do shluků. Proto je třeba prozkoumat vliv parametru σ na výsledek shlukování podrobněji.

3.4.3.1 Závislost výsledku shlukování na hodnotě σ

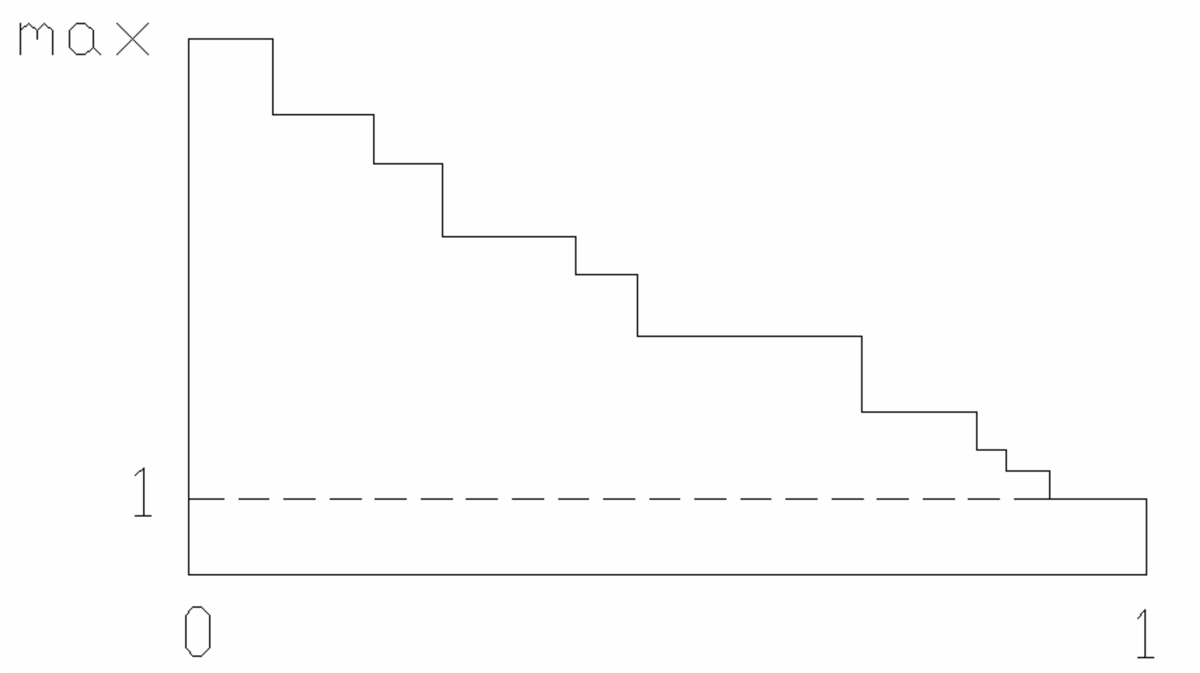
Parametr σ vystupuje ve vztahu pro výpočet funkce vlivu, tedy vzdálenosti, do jaké je objekt považován za sousední. Dvě nejpoužívanější funkce vlivu jsou obdélníková funkce vlivu:

$$f_{Square}=0 \text{ pro } d(i,j) > \sigma$$
$$f_{Square}=1 \text{ pro } d(i,j) \leq \sigma$$

a Gaussovská funkce vlivu:

$$f_{Gauss} = e^{-\frac{d(i,j)^2}{2\acute{o}^2}}$$

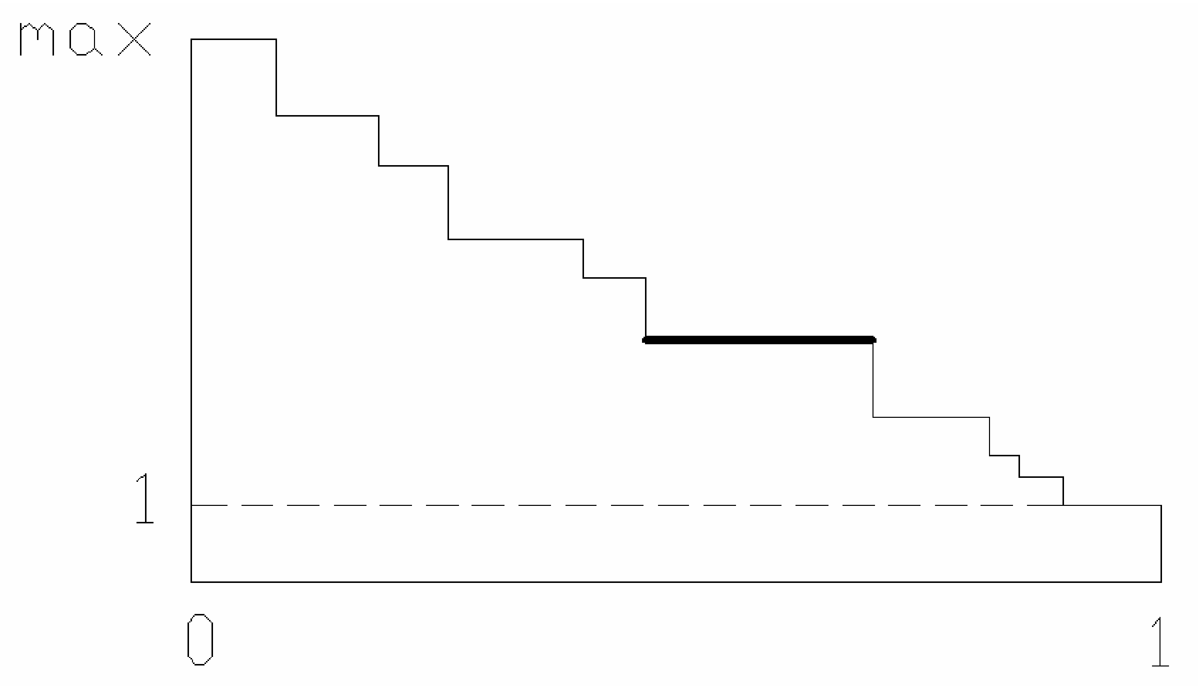
Je zřejmé, že při zvyšování hodnoty σ se zvyšuje hodnota funkce hustoty v jeho okolí, resp. zvětšuje se oblast, v níž má přítomnost objektu vliv na hodnotu funkce hustoty. Z toho vyplývá, že se zvyšující se hodnotou σ bude počet lokálních maxim této funkce, tedy počet nalezených shluků, klesat nebo zůstávat stejný, nikdy však nemůže vzrůst. Jinými slovy, funkce vyjadřující počet nalezených shluků v závislosti na hodnotě parametru σ je nerostoucí funkce, která nabývá pro σ=0 hodnoty *max*, která je rovna počtu objektů, a pro maximální hodnotu σ hodnoty 1, jak je vidět na následujícím obrázku:



na vodorovné ose je vynesena σ v normalizovaných hodnotách, na svislé výsledný počet shluků; *max* je počet objektů v prostoru dat; graf je pouze schematický, osa y není v lineárním měřítku

3.4.3.2 Nalezení nejvhodnější hodnoty parametru σ

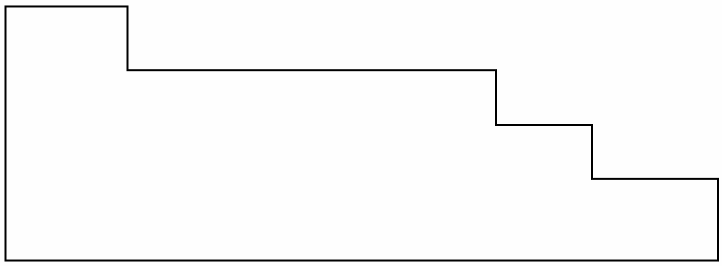
Na základě tohoto faktu se můžeme pokusit určit, jaká hodnota σ je nejvhodnější. Jak jsem již zmínil, obecně nelze rozhodnout, jaký výsledek shlukování je absolutně správný. Proto nelze ani určit absolutně správnou hodnotu σ. Za optimální rozdělení do shluků se dá považovat takové rozdělení, které je nejstabilnější ze všech možných. Jinými slovy takové, u něhož je třeba co největší změny vstupních parametrů, zde konkrétně parametru σ, aby se změnil výsledek. To v praxi znamená, že optimální hodnota σ by měla být vybrána z nejdelšího intervalu, ve kterém počet shluků zůstává stejný. Rozdělení provedené s takovou hodnotou parametru σ můžeme považovat za přirozené pro daný prostor dat, a tedy optimální [4].



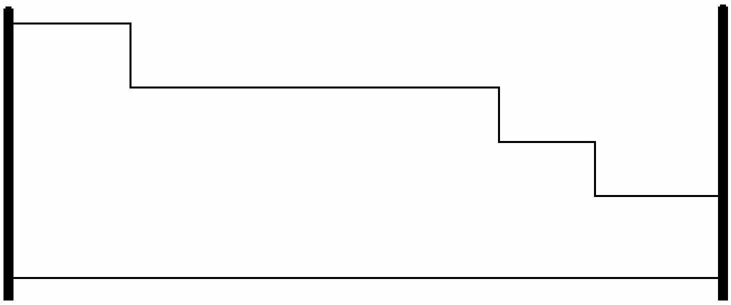
obrázek se zvýrazněným nejdelším intervalem neměnícího se počtu shluků, z něhož by měla být vybrána hodnota σ aby mohl být výsledek považován za pro daná vstupní data přirozené rozdělení do shluků

Implementace tohoto postupu tedy spočívá v tom, že shlukovou analýzu je třeba provést vícekrát pro různé hodnoty parametru σ, a to tolikrát, aby bylo možno identifikovat nejdelší interval, na němž nedochází ke změně počtu shluků. Vzhledem k tomu, že výpočetně nejnáročnější částí algoritmu je výpočet vzdálenosti objektů, a ta se nemění, nepřispívá opakované provádění shlukování zásadně k celkové výpočetní náročnosti programu.

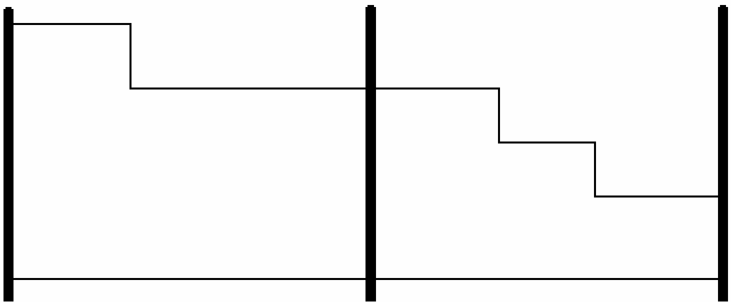
Ve své implementaci postupuji tak, že provedu shlukovou analýzu nejdříve pro minimální hodnotu σ a pak pro maximální. Následně rekurzivně provádím shlukovou analýzu pro hodnotu σ rovnu vždy polovině intervalu mezi už spočítanými výsledky (na polovinu zmenšuji krok, se kterým prohledávám interval mezi minimální hodnotou σ a maximální hodnotou σ). Po každém cyklu zjišťuji, jaký je nejdelší nalezený interval, a proceduru je možno ukončit, když je dvojnásobek kroku menší, než rozdíl nejdelšího a druhého nejdelšího nalezeného intervalu. Postup hledání nejdelšího intervalu lze znázornit těmito ilustracemi:



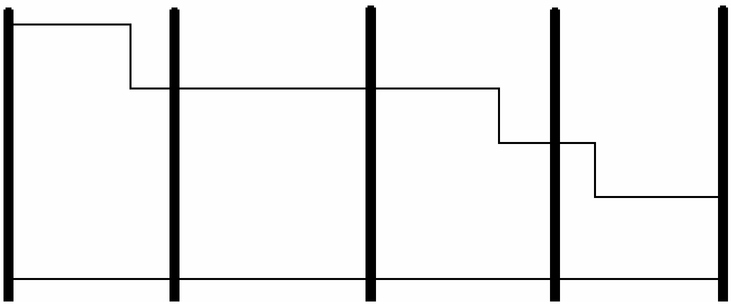
(1) zjednodušený příklad znázornění závislosti počtu shluků
na hodnotě parametru σ



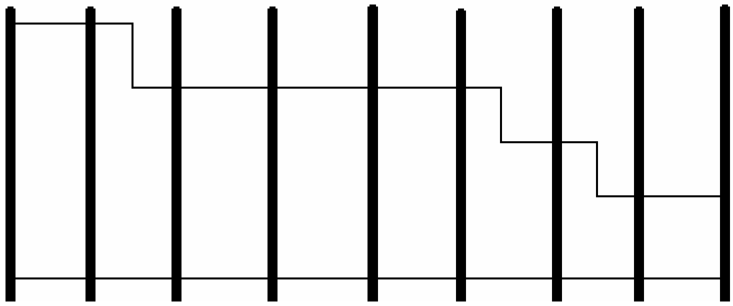
(2) v prvním kroku jsou zjištěny hodnoty na obou koncích intervalu



(2) v každém dalším kroku jsou zjištěny hodnoty vždy v polovině
intervalu mezi už známými; nejdříve pouze jedna



(3) ve třetím kroku je nalezena dvojice stejných hodnot počtu shluků,
protože však rozdíl nejdelšího a druhého nejdelšího intervalu (kterým
je v tomto případě kterýkoliv z ostatních nalezených) není větší než
dvojnásobek kroku (tedy počet nalezených bodů z nejdelšího intervalu
není o dvě větší než počet bodů z druhého nejdelšího), hledání pokračuje

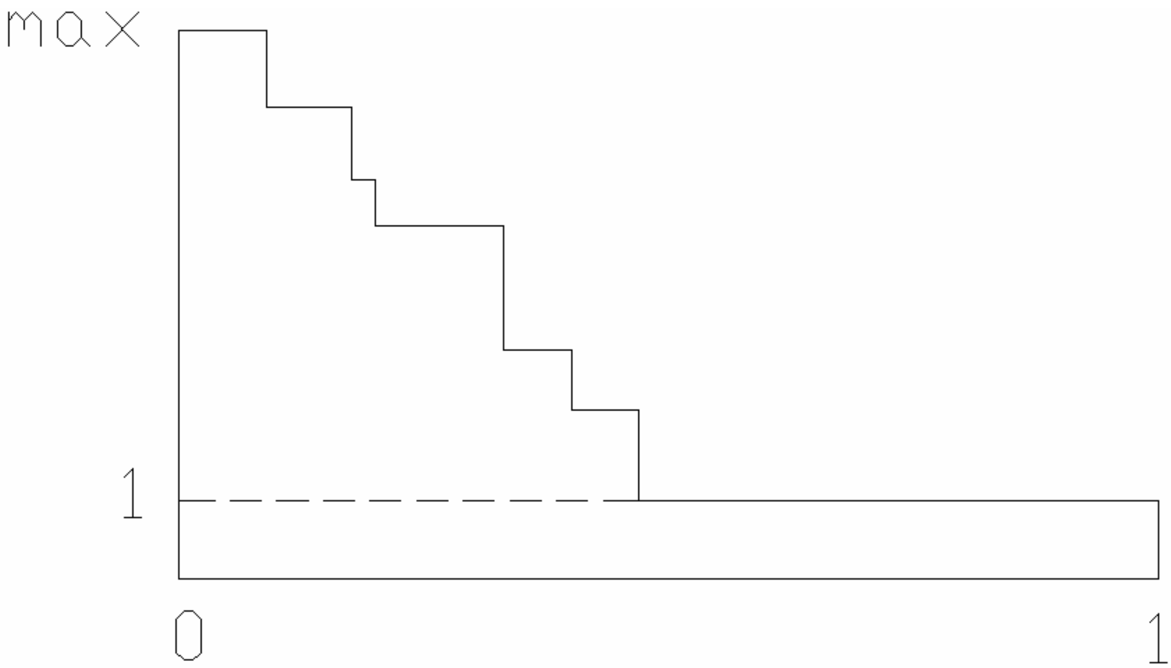


(4) ve čtvrtém kroku jsou nalezeny další dva body nejdelšího intervalu, z něhož tak nyní máme už čtyři body, zatímco do druhého nejdelšího intervalu (ať už je jím kterýkoliv z obou krajních) padly pouze body dva; je tedy zaručeno, že nemůžou být delší než první kandidát a algoritmus končí

Na závěr je ještě nutno vybrat, kterou konkrétní hodnotu z nalezeného intervalu použít. Zvolil jsem nejnižší hodnotu σ, tedy levý okraj nalezeného intervalu, neboť shluky jsou s použitím nižších hodnot σ kompaktnější; v praxi to má vliv na to, který objekt bude určen jako střed shluku, při použití nižší hodnoty σ je toto rozhodování méně ovlivněno případným vlivem objektů ze sousedních shluků.

3.4.3.3 Problémy spojené s automatickým určováním nejvhodnější hodnoty σ

Pokud stanovíme hodnotu parametru σ výše uvedeným způsobem, setkáme se s problémem stanovení maximální hodnoty σ, do kterých má ještě smysl zjišťovat počet shluků, do nichž budou objekty za jejího užití rozděleny. Nejde však jen o optimalizační důvody. Graf závislosti počtu shluků na hodnotě σ totiž – pokud využijeme celý rozsah možných hodnot σ, tedy interval <0;1> – ve skutečnosti vypadá často podobně jako na následujícím obrázku:



ve skutečnosti je závislost počtu shluků na hodnotě σ podobná tomuto schématu – nejdelším intervalem je interval poslední

Nejdelším intervalem, v němž se počet shluků nezmění, je interval končící hodnotou σ=1, a při použití hodnoty σ z tohoto intervalu bude výsledkem jediný shluk. To jistě není nejvhodnější řešení.

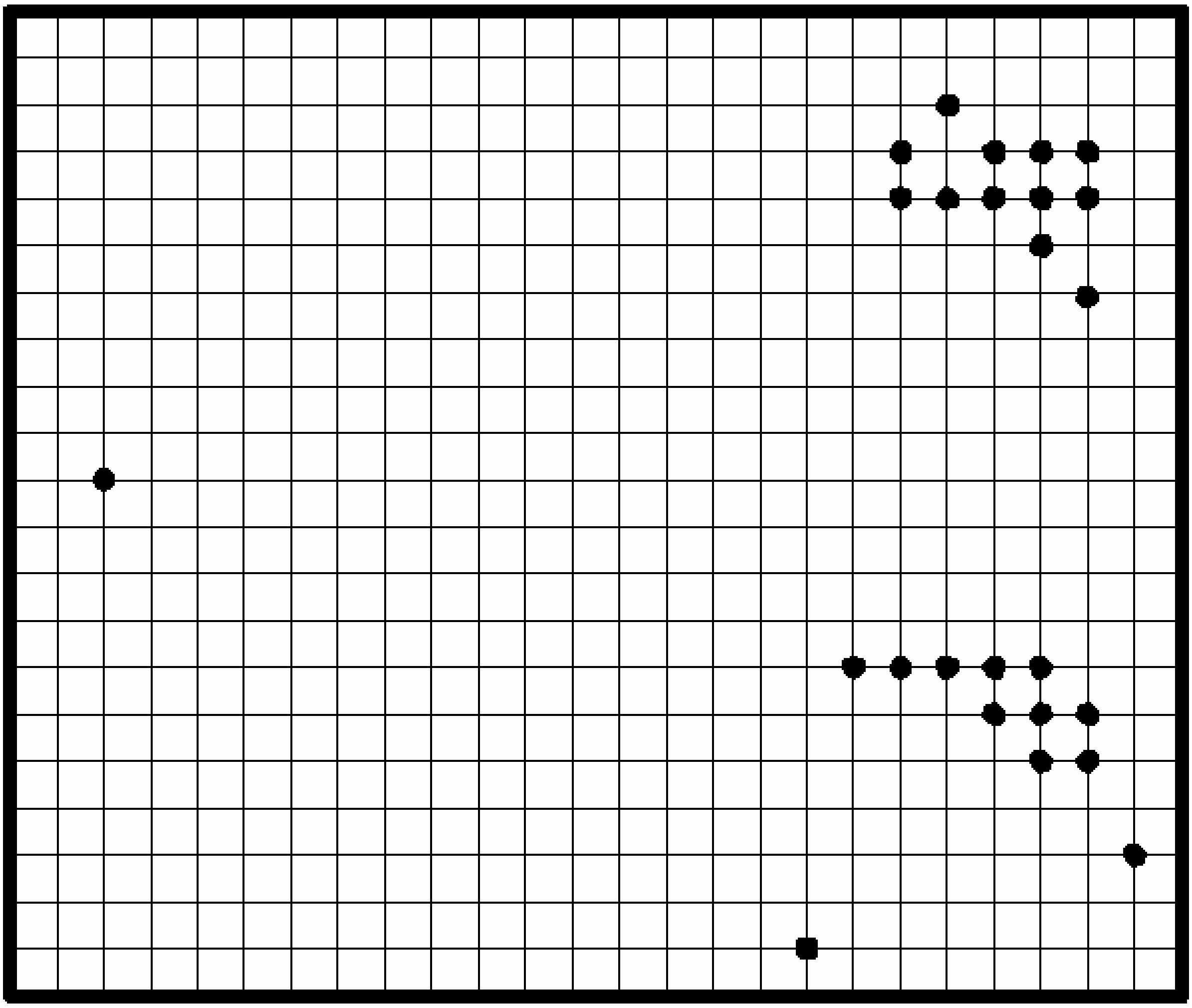
Prostou úvahou lze dospět k tomu, že takovýto výsledek je téměř jistý v případě, že hodnota σ je vyšší než 0,5, neboť v tom případě vlivová funkce všech objektů sahá do středu prostoru dat a objekt nacházející se mu nejblíže se tak stává středem jediného shluku, který bude nalezen. Prvním krokem je tedy omezení hodnoty σ na hodnoty nižší než 0,5.

Toto omezení však ještě nezaručí, že shluková analýza neskončí nalezením jednoho shluku obsahující všechny objekty. Pokud jsou data rozmístěna tak, že tvoří několik shluků, které však od sebe nejsou výrazně odděleny (vzdálenost mezi jednotlivými objekty ve shluku je relativně velká v poměru ke vzdálenosti mezi shluky), budou pravděpodobně na základě výše uvedeného kriteria nejdelšího intervalu, pro nějž se počet shluků nemění, sloučena do jediného shluku. V takovém případě by bylo možné se na takový výsledek dívat jako na korektní, neboť výše bylo uvedeno, že právě takovéto rozdělení lze pro dané rozložení objektů považovat za přirozené. Protože však uživatel pravděpodobně takovýto výsledek nebude považovat za přínosný, upravil jsem výše popsaný algoritmus hledání nejdelšího intervalu tak, aby interval odpovídající rozdělení objektů do jediného shluku ignoroval. Takovéhoto výsledku tudíž není možné dosáhnout, v tomto případě bude nalezen druhý nejdelší interval a hodnota σ bude vybrána z něj.

Opačným extrémem, ke kterému může dojít, je nalezení tolika shluků, kolik je na vstupu objektů. K tomu dochází podobným způsobem jako k předchozímu výsledku, a to tehdy, když shluky nejsou výrazně odděleny a rozmístění objektů v prostoru dat je relativně řídké (zatímco k nalezení jediného shluku dochází v případě, že objekty jsou rozmístěny relativně hustě). Zde platí to co u jednoho shluku – uživatel zřejmě nebude s takovýmto výsledkem spokojen, a proto je vhodné i tady donutit program nalézt druhý nejdelší interval.

Nakonec se ukazuje jako vhodné poskytnout uživateli možnost vkládat maximální a minimální hodnotu σ jako parametr programu. Takovýto parametr je pro uživatele sice dosti neprůhledný, ale prakticky se dá použít v případě, že výsledkem shlukování bude takový počet shluků, který bude uživatel považovat za příliš malý; pak prostě snížením hodnoty maximální σ může program donutit nalézt jiné rozdělení, s větším počtem shluků. Stejně lze postupovat i se zadáváním minimální hodnoty σ, kterou je možno ručně zvýšit v případě, že nalezený počet shluků uživateli připadá příliš velký. Manipulací s intervalem hodnot, kterých může nabývat σ, vlastně uživatel přiměje program nalézt nikoliv přirozené rozdělení objektů do shluků, ale „nejpřirozenější“ rozdělení z těch, které umožňuje zadaný interval hodnot σ.

Dále je nutno upozornit na jeden negativní důsledek popsaného algoritmu na nalezení nejvhodnější hodnoty parametru σ. Ačkoliv metody založené na hustotě obecně nemají žádný problém vypořádat se s odlehlými hodnotami, ve speciálních případech mohou mít odlehlé hodnoty velmi výrazný vliv na proces automatického nalezení nejvhodnější hodnoty parametru σ. Konkrétně jde o vstupní data, v nichž se odlehlé hodnoty nacházejí na okraji prostoru dat, tedy taková, u nichž některý z atributů nabývá hodnot významně menších/větších než tytéž atributy u objektů, které nejsou odlehlé (jsou součástí některého ze shluků). Taková situace je znázorněna na následujícím obrázku:



Ukázka rozložení vstupních dat, kdy jeden objekt má hodnotu jednoho z atributů (x-ovou souřadnici) výrazně vzdálenou od hodnot téhož atributu ostatních objektů

Důsledkem existence byť jediného objektu s hodnotou atributu takto vzdálenou hodnotám odpovídajících atributů ostatních objektů je deformace procesu hledání nejvhodnější hodnoty parametru σ, neboť existuje značná pravděpodobnost, že bude jako optimální nalezena hodnota parametru σ taková, která objekty rozdělí do dvou shluků, z nichž jeden bude tvořit onen odlehlý objekt a druhý ostatní objekty. Nelze přesně matematicky vyjádřit, kdy k tomuto případu dojde, neboť to záleží na rozložení vstupních dat, ale pravděpodobnost, že se tak stane, je tím vyšší, čím větší je poměr vzdálenosti odlehlého objektu od nejbližšího shluku ke vzdálenostem shluků navzájem. Tuto pravděpodobnost naopak snižuje počet dalších atributů (rozměrů prostorů dat) v nichž objekty s výrazně odlehlými hodnotami neexistují.

Takovýto výsledek je zcela přirozený a teoreticky i on je správný, neboť prostor dat je prostě určen minimálními a maximálními hodnotami jednotlivých parametrů a ty jsou v tomto případě od sebe mnohem výrazněji vzdáleny než hodnoty atributů ostatních objektů, které potom splňují podmínku pro vytvoření shluku, tedy malou vzdálenost jednotlivých objektů ve shluku oproti vzdálenosti objektů mimo shluk. Problém je spíše ve významu, který se odlehlým hodnotám přikládá

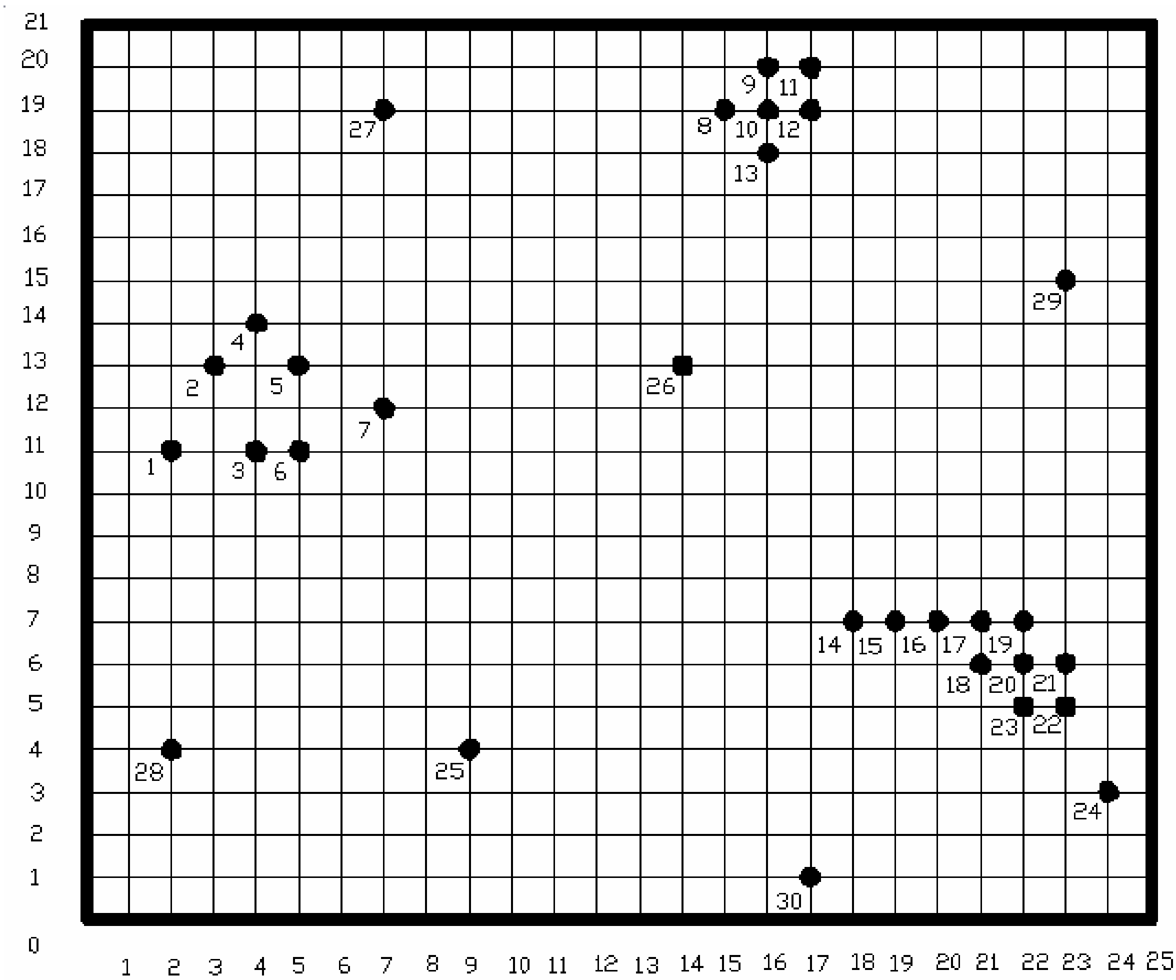
– většinou jsou považovány za šum a nebo neobvyklé hodnoty, které by neměly ovlivňovat výsledek shlukování. Je tedy vhodné odlehlé hodnoty takovéhoto typu ze shlukovaných dat předem ručně odstranit a provádět shlukování bez nich. Avšak pokud se tak nestane, je samozřejmě možno výsledek upravit dodatečným snížením maximální hodnoty σ.

Konečně je vhodné zamyslet se nad tím, co se stane, pokud budou na vstup vložena náhodná data. Taková data lze těžko nějak přirozeně rozdělit do shluků. Implementovaný algoritmus však nějaké rozdělení vždy najde. Může to být jeden extrém – jediný shluk, a nebo druhý extrém – tolik shluků, kolik je na vstupu objektů. Jak jsem však už psal výše, v těchto případech algoritmus hledá druhou nejlepší hodnotu σ (respektive třetí, pokud druhá je druhý extrém). Pokud jde o náhodné rozložení objektů na vstupu, často se stane, že je nalezeno více „nejdelších" intervalů, resp. více intervalů o téměř stejné délce. Pro tuto eventualitu je proto třeba do algoritmu pro hledání nejdelšího intervalu přidat dodatečnou podmínku, která algoritmus ukončí při dosažení určitého nejmenšího kroku, neboť jinak hrozí, že bude výpočet buď příliš dlouhý a nebo dokonce nekonečný. Je totiž zřejmé, že nemá smysl určovat, který z několika téměř stejně dlouhých intervalů je o nepatrný kousek delší, neboť to značí, že všechny výsledky založené na jednotlivých intervalech jsou prakticky stejně „přirozené". V případě že jde o shlukování náhodně rozložených objektů pak už vůbec nemá smysl se tím zabývat, neboť to je z principu bezvýsledné.

4 Ověření funkčnosti, testování

4.1 Názorný příklad

K základnímu ověření funkčnosti jsem zvolil příklad, z něhož je na první pohled patrné, jaký by měl být výsledek shlukování – jde o dvourozměrnou plochu s 20 body, rozmístěnými podle následujícího obrázku (vstupní soubor je obsahem přílohy č. 2):



Očekáváme, že by mělo být nalezeno 9 shluků, resp. tři skutečné shluky a šest odlehlých hodnot, které se projeví jako shluky tvořené jediným objektem. Výsledky programu vypadají takto:

Pocet nalezenych shluku: 9
Hodnota parametru sigma: 0.078125
----

stred: objekt5 - x: 5; y: 13;
pocet objektu: 6

stred: objekt8 - x: 15; y: 19;
pocet objektu: 6

stred: objekt17 - x: 21; y: 7;
pocet objektu: 11

stred: objekt25 - x: 9; y: 4;
pocet objektu: 1

stred: objekt26 - x: 14; y: 13;
pocet objektu: 1

stred: objekt27 - x: 7; y: 19;
pocet objektu: 1

stred: objekt28 - x: 2; y: 4;
pocet objektu: 1

stred: objekt29 - x: 23; y: 15;
pocet objektu: 1

stred: objekt30 - x: 17; y: 1;
pocet objektu: 1

Lze tedy konstatovat, že program našel přirozené rozdělení těchto dat do shluků, tak jak člověk intuitivně odvodí z pohledu na grafické znázornění vstupních dat.

4.2 Praktický příklad

Jako druhý příklad jsem vybral seznam světových států se dvěma atributy, a to rokem vzniku a kontinentem, na němž stát leží. Vstupní údaje jsou obsahem přílohy č. 3. Výsledkem shlukové analýzy bez omezení parametru σ je v tomto případě rozdělení do shluků podle kontinentů. To je způsobeno normalizací dat, kdy výsledný prostor dat je deformován tím, že na ose časové nabývají atributy hodnot od 1776 po 2008, kdežto na ose kontinentů, která díky normalizaci je stejně významná jako osa roků, nabývají jen pěti hodnot a jako přirozené se tudíž jeví rozdělení podle kontinentů. Toto vyplývá z podstaty nominálního typu atributu a je zřejmé, že s tímto typem atributu je třeba zacházet opatrně a s ohledem na tyto souvislosti.

V každém případě toto asi není příliš přínosný výsledek, a tak se tady uplatní možnost nastavit si horní hranici parametru σ – jestliže nám předchozí výsledek nevyhovuje a chtěli bychom provést rozdělení do více shluků, všimneme si, že byla použita hodnota parametru σ = 0.103125. Nastavíme tedy ručně maximální hodnotu sigmy na například 0.1 a dostaneme tyto výsledky:

```
Pocet nalezenych shluku: 10
Hodnota parametru sigma: 0.06875
-------------------------------------------

stred: Algeria - rok: 1962; kontinent: Afrika;
pocet objektu: 52

stred: Trinidad and Tobago - rok: 1962; kontinent: Amerika;
pocet objektu: 15

stred: Argentina - rok: 1816; kontinent: Amerika;
pocet objektu: 19

stred: Syria - rok: 1961; kontinent: Asie;
pocet objektu: 37

stred: Thailand - rok: 1776; kontinent: Asie;
pocet objektu: 1

stred: Albania - rok: 1913; kontinent: Evropa;
pocet objektu: 11

stred: Netherlands - rok: 1839; kontinent: Evropa;
pocet objektu: 9

stred: Belarus - rok: 1990; kontinent: Evropa;
pocet objektu: 14

stred: Australia - rok: 1901; kontinent: Oceanie;
pocet objektu: 1
```

stred: Fiji - rok: 1970; kontinent: Oceanie;
pocet objektu: 12

Výsledné rozdělení už velmi dobře odpovídá historickým souvislostem. Vidíme jeden, a to ten největší, shluk zahrnující africké státy, neboť ty získaly svou nezávislost téměř všechny v 60. a 70. letech 20. století, americké státy jsou rozděleny do dvou shluků, jeden zahrnující vlnu vzniku nezávislých států na počátku 18. století, druhý novější (většinou státy tzv. Západní Indie, které získaly nezávislost v průběhu druhé poloviny 20. století). Asijské státy získaly svou nezávislost zpravidla ve 2. polovině 20. století, ale shluková analýza identifikovala výjimku, Thajsko, které vzniklo už roku 1776 a tvoří tak odlehlou hodnotu. Podobnou výjimku ve své oblasti tvoří i Austrálie, neboť ta je nezávislá od roku 1901, kdežto ostatní státy v Oceánii vznikly až ve druhé polovině 20. století. A konečně v Evropě byly identifikovány tři shluky, odpovídající třem fázím výraznějších změn na tomto kontinentě, a to období kolem poloviny 19. století, období kolem první světové války, a období po rozpadu SSSR a bývalé Jugoslávie.

Závěr

Tato práce demonstruje funkčnost metody DENCLUE pro shlukovou analýzu. Pokusil jsem se navrhnout postup řešení problému zadávání základního parametru metody, hodnoty σ. Ukázalo se, že tzv. přirozené rozdělování do shluků, založené na nejdelším intervalu, v němž se počet shluků se zvyšující se hodnotou σ nemění, funguje teoreticky velmi dobře (jak ukazuje názorný příklad č. 1), ale v praxi po shlukové analýze často neočekáváme toto teoreticky optimální řešení. Je proto vhodné umožnit uživateli do procesu výběru hodnoty σ aktivně zasáhnout, i když nemusí teorii za ní zcela rozumět.

V praxi by bylo vhodné buď aplikaci navrhnout tak, aby rovnou nabízela několik variant výsledků (tedy nejen výsledek za použití hodnoty σ z nejdelšího intervalu beze změny počtu shluků, ale více výsledků z několika nejdelších intervalů), a nebo umožnit uživateli, aby po shlédnutí výsledku prostě zadal, že chce prostor dat rozdělit na větší počet shluků (případně menší) a novou maximální (případně minimální) hodnotu σ stanovit na základě předchozí automaticky. V této ukázkové aplikaci jsem ponechal ruční zadávání hodnoty σ, aby bylo možno názorně ukázat, jak ovlivňuje výsledek a jak se s ní pracuje.

Hlavním přínosem způsobu určování optimální hodnoty σ popsaného v této práci je fakt, že po uživateli nevyžaduje žádné zadávání číselných hodnot, jejichž smyslu by nerozuměl, nýbrž pokusí se nalézt přirozené rozdělení objektů do shluků, a pokud toto uživateli nevyhovuje, ten jednoduše stanoví, zda chce nalézt větší (snížení maximální hodnoty σ pod optimum) nebo menší (zvýšení minimální hodnoty σ nad optimum) počet shluků.

Literatura

[1] A Tutorial on Clustering Algorithms. Dokument dostupný na URL: http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/index.html (duben 2008)

[2] Zendulka, J. a kol.: Získávání znalostí z databází - studijní opora. FIT VUT Brno, 2006.

[3] Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers. ISBN 1-55860-901-3.

[4] Hinneburg A., Keim D.A.: „An Efficient Approach to Clustering in Large Multimedia Databases with Noise“, Proc. 4rd Int. Conf. on Knowledge Discovery and Data Mining, AAAI Press, 1998. Dokument dostupný na URL: http://infovis.uni-konstanz.de/papers/1998/kdd98.pdf (duben 2008)

Seznam příloh

Příloha 1:
Formát souboru se vstupními daty

Soubor se vstupními daty je textový soubor v následujícím formátu:

	název_atributu_1	název_atributu_2	…	název_atributu_*n*
	typ_atributu_1	typ_atributu_2	…	typ_atributu_*n*
objekt_1	hodnota_11	hodnota_12	…	hodnota_1*n*
objekt_2	hodnota_21	hodnota_22	…	hodnota_2*n*
…	…	…	…	…
objekt_*n*	hodnota_*n*1	hodnota_*n*2	…	hodnota_*nn*

<CR><LF>

kde:

- název_atributu je řetězec identifikující atribut (může obsahovat mezery a nemusí být unikátní; slouží k zpřehlednění výsledného výpisu)
- typ_atributu určuje typ proměnné a může nabývat jedné z těchto hodnot:
  - interval – intervalová proměnná (např. 14, 126.47…)
  - binarni – binární proměnná (0 nebo 1)
  - binarni_asym – binární proměnná, u níž je kladná hodnota významnější než záporná
  - nominalni – nominální proměnná, nabývající libovolných (i textových) hodnot (např. málo, hodně, středně)
- objekt je řetězec identifikující atribut (může obsahovat mezery a nemusí být unikátní; slouží k zpřehlednění výsledného výpisu)
- hodnota je hodnota daného atributu daného objektu, odpovídající typu (viz. výše)
- <CR><LF> je prázdný řádek, značící konec vstupních dat; případný zbytek souboru se nebere v potaz

Příklad vstupního souboru:

	souřadnice x	souřadnice y
	interval	interval
bod1	6	14
bod2	14	3
bod4	12.6	17
bod5	2.7	4
bod6	1.6	76
bod7	16.45	9
bod8	0.3	16

Příloha 2:

Soubor vstupních dat pro příklad č. 1

	x	y
	interval	interval
objekt1	2	11
objekt2	3	13
objekt3	4	11
objekt4	4	14
objekt5	5	13
objekt6	5	11
objekt7	7	12
objekt8	15	19
objekt9	16	20
objekt10	16	19
objekt11	17	20
objekt12	17	19
objekt13	16	18
objekt14	18	7
objekt15	19	7
objekt16	20	7
objekt17	21	7
objekt18	21	6
objekt19	22	7
objekt20	22	6
objekt21	23	6
objekt22	23	5
objekt23	22	5
objekt24	24	3
objekt25	9	4
objekt26	14	13
objekt27	7	19
objekt28	2	4
objekt29	23	15
objekt30	17	1

Příloha 3:
Soubor vstupních dat pro příklad č. 2

	rok	kontinent
	interval	nominalni
Algeria	1962	Afrika
Angola	1975	Afrika
Benin	1960	Afrika
Botswana	1966	Afrika
Burkina Faso	1960	Afrika
Burundi	1962	Afrika
Cameroon	1960	Afrika
Cape Verde	1975	Afrika
Central African Republic	1960	Afrika
Chad	1960	Afrika
Comoros	1975	Afrika
Congo, DR	1960	Afrika
Congo, Rep.	1960	Afrika
Côte d'Ivoire	1960	Afrika
Djibouti	1977	Afrika
Equatorial Guinea	1968	Afrika
Eritrea	1993	Afrika
Ethiopia	1941	Afrika
Gabon	1960	Afrika
Gambia	1965	Afrika
Ghana	1957	Afrika
Guinea	1958	Afrika
Guinea-Bissau	1974	Afrika
Kenya	1963	Afrika
Lesotho	1966	Afrika
Libya	1951	Afrika
Madagascar	1960	Afrika
Malawi	1964	Afrika
Mali	1960	Afrika
Mauritania	1960	Afrika
Mauritius	1968	Afrika
Morocco	1956	Afrika
Mozambique	1975	Afrika
Namibia	1990	Afrika
Niger	1960	Afrika
Nigeria	1960	Afrika
Rwanda	1962	Afrika
Sao Tomé and Príncipe	1975	Afrika
Senegal	1960	Afrika

Seychelles	1976	Afrika
Sierra Leone	1961	Afrika
Somalia	1960	Afrika
South Africa	1931	Afrika
Sudan	1956	Afrika
Swaziland	1968	Afrika
Tanzania	1961	Afrika
Togo	1960	Afrika
Tunisia	1956	Afrika
Uganda	1962	Afrika
Zambia	1964	Afrika
Zimbabwe	1980	Afrika
Antigua and Barbuda	1981	Amerika
Argentina	1816	Amerika
Bahamas	1973	Amerika
Barbados	1966	Amerika
Belize	1981	Amerika
Bolivia	1825	Amerika
Brazil	1825	Amerika
Canada	1931	Amerika
Chile	1818	Amerika
Colombia	1810	Amerika
Costa Rica	1821	Amerika
Cuba	1868	Amerika
Dominica	1978	Amerika
Dominican Republic	1865	Amerika
Ecuador	1822	Amerika
El Salvador	1821	Amerika
Grenada	1974	Amerika
Guatemala	1821	Amerika
Guyana	1966	Amerika
Haiti	1844	Amerika
Honduras	1821	Amerika
Jamaica	1962	Amerika
Mexico	1810	Amerika
Nicaragua	1821	Amerika
Panama	1903	Amerika
Paraguay	1811	Amerika
Peru	1821	Amerika
Saint Kitts and Nevis	1967	Amerika
Saint Lucia	1979	Amerika
Saint Vincent and the Grenadines	1979	Amerika
Suriname	1975	Amerika
Trinidad and Tobago	1962	Amerika
United States	1776	Amerika
Uruguay	1830	Amerika

Afghanistan	1919	Asie
Armenia	1990	Asie
Azerbaijan	1991	Asie
Bahrain	1971	Asie
Bangladesh	1971	Asie
Bhutan	1885	Asie
Brunei	1984	Asie
Cambodia	1989	Asie
Georgia	1991	Asie
India	1947	Asie
Indonesia	1945	Asie
Iraq	1932	Asie
Israel	1948	Asie
Jordan	1946	Asie
Kazakhstan	1991	Asie
Korea, North	1945	Asie
Korea, South	1945	Asie
Kuwait	1969	Asie
Kyrgyzstan	1991	Asie
Laos	1949	Asie
Lebanon	1941	Asie
Malaysia	1957	Asie
Maldives	1965	Asie
Mongolia	1911	Asie
Myanmar	1948	Asie
Pakistan	1947	Asie
Philippines	1946	Asie
Qatar	1971	Asie
Saudi Arabia	1927	Asie
Singapore	1959	Asie
Sri Lanka	1948	Asie
Syria	1961	Asie
Taiwan	1949	Asie
Tajikistan	1991	Asie
Thailand	1776	Asie
Timor-Leste	1975	Asie
Turkmenistán	1991	Asie
United Arab Emirates	1971	Asie
Uzbekistan	1991	Asie
Albania	1913	Evropa
Andorra	1813	Evropa
Belarus	1990	Evropa
Belgium	1830	Evropa
Bosnia and Hercegovina	1992	Evropa
Bulgaria	1908	Evropa
Croatia	1991	Evropa

Cyprus	1960	Evropa
Czech Republic	1918	Evropa
Estonia	1991	Evropa
Finland	1917	Evropa
Germany	1870	Evropa
Greece	1829	Evropa
Iceland	1944	Evropa
Ireland	1922	Evropa
Italy	1870	Evropa
Kosovo	2008	Evropa
Lithuania	1990	Evropa
Latvia	1990	Evropa
Liechtenstein	1813	Evropa
Luxembourg	1839	Evropa
Macedonia	1991	Evropa
Malta	1964	Evropa
Moldova	1991	Evropa
Monaco	1861	Evropa
Montenegro	2006	Evropa
Netherlands	1839	Evropa
Norway	1905	Evropa
Poland	1918	Evropa
Romania	1877	Evropa
Serbia	1918	Evropa
Slovakia	1918	Evropa
Slovenia	1991	Evropa
Ukraine	1991	Evropa
Australia	1901	Oceanie
Fiji	1970	Oceanie
Kiribati	1979	Oceanie
Marshall Islands	1979	Oceanie
Micronesia	1979	Oceanie
Nauru	1968	Oceanie
Palau	1981	Oceanie
Papua New Guinea	1975	Oceanie
Samoa	1962	Oceanie
Solomon Islands	1978	Oceanie
Tonga	1970	Oceanie
Tuvalu	1978	Oceanie
Vanuatu	1980	Oceanie
Egypt	1922	Afrika