Strojové učenie v systémoch spracovania informácií

Kristína Machová

Košice, 2010

Doc. Ing. Kristína Machová, PhD.
Katedra kybernetiky a umelej inteligencie
Fakulta elektrotechniky a informatiky
Technická univerzita v Košiciach
Kristina.machova@tuke.sk

Monografia vznikla za podpory VEGA grantu MŠ SR č. 1/4074/07 „Metódy anotovania, vyhľadávania, tvorby a sprístupňovania znalostí s využitím meta - dát pre sémantický popis znalostí“ a taktiež za podpory VEGA grantu MŠ SR č. 1/0076/10 „Inteligentné vyhľadávanie informácií na webe obohatené o sémantický aspekt“ a VEGA grantu MŠ SR č. 1/0042/10 „Metódy identifikácie, anotovania, vyhľadávania, sprístupňovania a kompozície služieb s využitím sémantických metadát pre podporu vybraných typov procesov“.

Edícia vedeckých spisov Fakulty elektrotechniky a informatiky TU Košice.
Lektorovali: *prof. Ing. Igor Mokriš, CSc.*
doc. Ing. Ján Paralič, PhD.
doc. Ing. Marián Mach, CSc.

ISBN

Alebo všetko zisti naisto,
alebo to pusť z hlavy.
Pretože záver, ku ktorému dôjdeš,
môže byť iba tvoj,
a nie práve ten pravý.

James Thurber

Predslov

Predkladaná kniha podáva problematiku princípov a aplikácií strojového učenia v systémoch spracovania informácií z textových dokumentov. Táto problematika je veľmi aktuálna najmä v súvislosti s problematikou dolovania v obsahu prezentácií webového priestoru. Nadväzuje na staršiu publikáciu autorky s názvom „Strojové učenie. Princípy a algoritmy."

Druhá kapitola predkladanej publikácie je venovaná práve základným princípom algoritmov strojového učenia a výpočtovej teórii strojového učenia. Sústreďuje sa na algoritmy strojového učenia najčastejšie používané v oblasti spracovania textových dokumentov.

Tretia kapitola je venovaná problematike zloženej klasifikácie pomocou metód „boosting" a „bagging", ktorá zvyšuje efektívnosť klasifikácie textových dokumentov slabým klasifikátorom.

Štvrtá kapitola sa sústreďuje na samotné spracovanie textových dokumentov, získaných z webu. Táto oblasť je v súčasnosti predmetom intenzívneho výskumu popredných vedeckých inštitúcií sveta. Jej cieľom je znalostný prístup k získavaniu informácií z webu, ktorý predstavuje obrovský celosvetový informačný priestor, manuálne nezvládnuteľný, ako sa konštatuje v posudku predkladanej práce profesora Ing. Igora Mokriša, CSc. V rozpore s pozornosťou, ktorá je tejto oblasti venovaná v cudzojazyčnej literatúre, iba sporadicky natrafí čitateľ na podobnú publikáciu v slovenskom jazyku. Cieľom predkladanej knihy je vyplniť túto medzeru a podať slovenským čitateľom informácie o tejto zaujímavej oblasti.

Práca sa nesnaží byť vyčerpávajúcim textom z predkladanej oblasti a nerobí si ani nárok na úplnosť. Je vhodná pre tých, ktorí chcú získať základný prehľad oblasti, ale aj pre tých, ktorí majú hlbší záujem o prezentovanú problematiku a detaily experimentov, na ktorých participovala autor-

ka publikácie. Citujem z posudku prof. Ing. Igora Mokriša, CSc.: „...predložená publikácia predstavuje ucelené dielo, ktoré svojim obsahom a metodologickým spracovaním má charakter vedeckej monografie a je vhodná ako základná literatúra na výučbu problematiky strojového učenia v systémoch spracovania informácií na princípoch umelej inteligencie na technických univerzitách. Je vhodná tiež pre ďalších odborníkov, pedagogických pracovníkov a študentov, pracujúcich v danej oblasti nielen teoreticky, ale aj v praxi. Predkladaná publikácia tým vyplňuje medzeru, ktorá v oblasti informatickej literatúry v našich podmienkach neustále trvá a ktorá sa stále citeľne prejavuje ako nedostatok lekčných fondov vo vzdelávacom procese v pôsobnosti vedeckej, technickej a vysokoškolskej komunity Slovenska.“

Predkladaná kniha vznikla na Katedre kybernetiky a umelej inteligencie Fakulty elektrotechniky a informatiky Technickej univerzity v Košiciach. Pri jej zostavovaní vychádzala autorka zo skúseností, získaných počas riešenia viacerých výskumných projektov a zabezpečovania výuky predmetu Strojové učenie.

Moja vďaka patrí recenzentom prof. Ing. Igorovi Mokrišovi, CSc., doc. Ing. Jánovi Paraličovi, PhD. a doc. Ing. Mariánovi Machovi, CSc. za starostlivé prečítanie rukopisu, opravu mnohých formálnych aj vecných chýb ale hlavne za podnetné návrhy, ktoré obohatili obsah predkladaného diela a ktoré boli v konečnej verzii textu zohľadnené. Ďakujem mojim kolegom z Katedry kybernetiky a umelej inteligencie FEI TU v Košiciach za cenné rady a pripomienky, podnety a podporu počas môjho pôsobenia na katedre. Tento text venujem mojim najbližším za pochopenie a toleranciu k mojej práci.

V Košiciach autorka

Obsah

1 ÚVOD

Učenie je neodmysliteľnou súčasťou nášho života. Keby sme sa z konkrétneho nešpecifikovaného dôvodu nemohli ďalej učiť nové poznatky, náš život by sa stal mimoriadne zložitým a potrebovali by sme, aby sa o nás niekto staral. Stali by sme sa v určitom zmysle invalidmi, ktorí si svoje barličky nosia v hlave. Zvlášť to platí v dnešnej dobe, keď sa pravidlá a zákony rýchlo menia, keď takmer každodenne pribúdajú nové technológie a nové možnosti. Preto nie je udivujúce, že vznikla technická disciplína – strojové učenie, ktorá sa snaží čiastočne simulovať túto dôležitú schopnosť človeka. Stále v tejto disciplíne pribúdajú nové poznatky. Tieto sa týkajú hlavne základných princípov strojového učenia, výpočtovej teórie strojového učenia, možností kombinovať klasifikátory založené na základných metódach strojového učenia za účelom optimalizácie výsledkov klasifikácie (bagging a boosting). Zaujímavé je aj použitie klasifikačných a zhlukovacích metód strojového učenia v takých oblastiach, ako je spracovanie textových dokumentov, aktívne učenie a generovanie kľúčových slov z textov.

V princípe je možné strojové učenie deliť na deduktívne a induktívne. Podobne ako človek, aj stroj sa môže učiť deduktívne, teda aplikovaním všeobecnejšieho princípu na špecifický problém. Príkladom takého prístupu je metóda učenia založeného na vysvetľovaní, pri ktorom sa z veľmi rozsiahlej doménovej teórie generuje špecifickejšia doména. Experimenty autorky s týmto druhom učenia sú uvedené v [Machová, 1995b]. Spomenutá špecifickejšia doména môže mať napríklad formu acyklického stromu, ktorý úplne postačuje na riešenie danej úlohy, resp. triedy úloh. Aplikácia metódy učenia založeného na vysvetľovaní na oblasť prenatálnej medicíny je opísaná v [Machová-Janovský, 2001].

S odstupom rokov sa ukázalo, že oveľa praktickejšie je simulovať induktívny prístup k učeniu. Teda prístup, pri ktorom sa z veľkého množstva jednoduchých pozorovaní na najnižšej úrovni všeobecnosti (z trénovacej množiny) vygenerujú závislosti, vzťahy, teda nové doteraz neznáme znalosti na vyššej úrovni všeobecnosti. Takýmto spôsobom sa človek učí od najrannejšieho detstva pravidlá, podľa ktorých je riadený najprv malý svet rodiny, neskôr svet školy a napokon svet dospelých ľudí. Induktívne učiace sa algoritmy s rôznou reprezentáciou výsledkov (najznámejšie sú algo-

ritmy generujúce klasifikačné pravidlá, rozhodovacie stromy, rozhodovacie zoznamy, naivný Bayesov klasifikátor a podobne) sa dodnes s úspechom používajú v rôznych oblastiach. Dá sa predpokladať, že tento vývoj bol ovplyvnený skutočnosťou, že v dnešnej dobe je prístupné veľké množstvo databáz a kolekcií dát či textových dokumentov, vytváraných rôznymi vzdelávacími inštitúciami ako aj komerčnými spoločnosťami. Komerčné spoločnosti (napríklad obchodné reťazce) sú niekedy ochotné sprístupniť aj databázy, ktoré si dovtedy chránili, ak vidia perspektívu získania znalostí, ktoré im pomôžu realizovať rozhodnutia, ktoré im zabezpečia v budúcnosti prosperitu. To sa deje v oblasti nazvanej dolovanie v dátach „data mining“, ktorej základom sú práve algoritmy strojového učenia.

V rámci skupiny induktívnych algoritmov strojového učenia sa uskutočňuje ešte delenie na kontrolované a nekontrolované učenie. Kontrolované učenie je učenie z označkovaných, teda vopred klasifikovaných dát (trénovacích príkladov). Skupina algoritmov, ktoré uskutočňujú kontrolované učenie sa nazýva „klasifikačné algoritmy“. O nekontrolovanom učení hovoríme, keď nemáme informácie o triedach trénovacích príkladov. Skupina algoritmov, ktoré uskutočňujú nekontrolované učenie (v rámci riešenia klasifikačnej úlohy) sa nazýva „zhlukovacie algoritmy“.

Druhá kapitola obsahuje v úvode základné princípy algoritmov strojového učenia. Totiž pri porovnávaní celej škály algoritmov strojového učenia sa začínajú rysovať spoločné črty určitých skupín algoritmov, teda základné princípy týchto algoritmov, ktoré sa dajú využiť pri navrhovaní nových algoritmov, vhodných pre riešenie konkrétnej triedy úloh. Taktiež tu vyvstáva otázka vypočítateľnosti, minimálnej početnosti trénovacej množiny, ktorá zabezpečí dobré výsledky v rámci zadanej tolerancie chýb, otázka nakoľko je získaná hypotéza dobrou aproximáciou cieľa. Odpovede na tieto otázky dáva výpočtová teória strojového učenia. Ďalej nasleduje prehľad najznámejších algoritmov strojového učenia, ktoré sú podrobnejšie popísané aj v učebnom texte „Strojové učenie. Princípy a algoritmy“ [Machová, 2002]. Potom nasleduje podkapitola, ktorá sa zameriava na tie algoritmy strojového učenia, ktoré sa typicky používajú pri spracovaní textových dokumentov. Druhá kapitola sa končí podkapitolou o odhade chyby klasifikácie od ktorého sa odvíja hodnotenie efektívnosti klasifikácie.

Ďalej je zaujímavé venovať sa otázke, či je možné danú presnosť a návratnosť klasifikačného algoritmu strojového učenia zlepšiť modifiká-

ciou trénovacej množiny a skladaním výsledkov mnohých klasifikácií, takzvaným hlasovaním. Ak áno, je zaujímavé nájsť odpoveď na otázku, koľko partikulárnych klasifikátorov musíme minimálne použiť, aby sme zvýšili presnosť a návratnosť klasifikácie. Odpovede na tieto otázky dávajú experimenty s metódami zloženej klasifikácie „bagging“ a „boosting“ uvedené v tretej kapitole. Tieto experimenty súvisia s oblasťou spracovávania textových dokumentov, keďže metódy zloženej klasifikácie boli overované práve na veľkých kolekciách textových dokumentov.

V rámci spracovávania textových dokumentov (kapitola 4) je potrebné začať možnými formami reprezentácie dokumentov. Ďalej sa táto kniha venuje predspracovaniu textových dokumentov a problematike váhovania termov, teda slov v dokumente. Dnes existuje veľké množstvo váhovacích schém, ktoré sa zameriavajú na určenie selektívnej sily termu. Najvyššiu selektívnu silu má term, ktorého frekvencia výskytu v konkrétnom dokumente je veľmi vysoká a zároveň je jeho výskyt v ostatných dokumentoch kolekcie veľmi zriedkavý. Najnižšiu selektívnu silu má term, ktorý sa pravidelne a rovnako často vyskytuje vo všetkých dokumentoch kolekcie. Váhovacie schémy je možné s úspechom použiť na generovanie kľúčových slov dokumentu, čo je obsahom nasledujúcej podkapitoly. Svoju nezastupiteľnú úlohu pri spracovávaní textov majú aj zhlukovacie algoritmy strojového učenia. V tejto práci sa kladie dôraz na takzvané zhlukovanie s kontrolovanou inicializáciou, ktoré môže zvýšiť efektívnosť spracovávania dokumentov. Za predpokladu, že je k dispozícii početná množina neoznačkovaných trénovacích príkladov, môžeme použiť zhlukovacie algoritmy s rôznym úspechom, ktorý závisí od konkrétnej použitej techniky a vhodnosti jej voľby pre riešenie daného problému. Taktiež je efektívnosť závislá od toho, či máme dopredu daný počet zhlukov, alebo sa tento počet prispôsobuje rozdeleniu trénovacej množiny. Dopredu stanovený počet zhlukov ešte pred začatím učenia môže viesť k skresleniu výsledkov. V takom prípade je vhodné aplikovať kontrolovanú inicializáciu. Pre každý zo stanoveného počtu zhlukov sa manuálne vyberie jeden typický trénovací príklad takzvané centrum. Nové zhluky sa budú zoskupovať okolo týchto centier, vybratých kontrolovanou inicializáciou. Testy potvrdili efektívnosť tohto postupu. Išlo o testy vykonávané nad veľkými kolekciami dokumentov.

Do oblasti experimentov s algoritmami strojového učenia pri spracovaní textových dokumentov patrí aj aktívne učenie, ktoré využíva konkrétne klasifikačné algoritmy. Technika aktívneho učenia sa môže s úspechom použiť, keď je síce k dispozícii veľký objem trénovacích príkladov, ale iba veľmi malá časť z nich je označkovaná, teda obsahuje aj informáciu o triede, do ktorej trénovací príklad patrí. Teda klasifikačné metódy je možné použiť iba nad malou časťou trénovacej množiny. Klasifikačné pravidlá vygenerované z tejto množiny sa následne použijú na klasifikáciu zatiaľ neoznačkovaných trénovacích príkladov. Tým sa získa nová kvalitnejšia a početnejšia trénovacia množina, z ktorej sa vygenerujú nové klasifikačné pravidlá s vyššou presnosťou klasifikácie, ako ukázali experimenty v závere štvrtej kapitoly.

2 STROJOVÉ UČENIE

Táto podkapitola nadväzuje na prácu „Strojové učenie. Princípy a algoritmy“ [Machová, 2002], v ktorej sú definované základné pojmy oblasti strojového učenia. V aktuálnej kapitole predkladanej knihy sú zhrnuté iba tie poznatky z oblasti strojového učenia, ktoré sú potrebné na pochopenie nasledovných kapitol o zloženej klasifikácii a o spracovaní textových dokumentov. Postupne budú nasledovať témy ako: základné princípy strojového učenia, výpočtová teória, algoritmy strojového učenia najčastejšie používané v spracovaní textových dokumentov a napokon hodnotenie efektívnosti klasifikácie.

2.1 Základné princípy algoritmov strojového učenia

V praxi sa k riešeniu učiacej úlohy pristupuje tak, že sa vyskúša zvolená skupina algoritmov, ktorá sa vyberie viac-menej intuitívne. Avšak je možný aj iný prístup, ktorý využíva existujúce súvislosti medzi základnými princípmi učiacich algoritmov a charakteristikami učiacich úloh [Machová, 2003].

Tab.1: Matica algoritmov strojového učenia

	klasifikačná úloha	sekvenčná úloha
kontrolované učenie – s učiteľom	**kontrolovaná klasifikácia**	učenie učňov
nekontrolované učenie – bez učiteľa	zhlukovanie	učenie odmenou a trestom

Pri formulácii základných princípov učiacich algoritmov boli uvažované algoritmy kontrolovaného učenia, určené na riešenie klasifikačnej úlohy. Táto skupina je v tab.1 označená ako kontrolovaná klasifikácia. Boli uvažované nasledovné algoritmy kontrolovanej klasifikácie, pričom v zátvorke je uvedená reprezentácia výstupu, ktorý generuje príslušný algoritmus:

VSS – Version Space Search (logické konjunkcie),
EGS – Exhaustive General to Specific (logické konjunkcie),
ESG – Exhaustive Specific to General (logické konjunkcie),
HGS – Heuristic General to Specific (logické konjunkcie),
HSG – Heuristic Specific to General (logické konjunkcie),
AQ11 – (disjunktívna normálna forma - DNS),
NSC – Nonincremental Separate and Conquer (DNS),
ID3 – Iterative Dichotomizer 3 (rozhodovacie stromy),
ID5R – Iterative Dichotomizer 5 Recursive (rozhodovacie stromy),
C4.5 – (rozhodovacie stromy),
MDLP – Min. Description Lenght Principle (rozhodovacie stromy),
CN2 – Clark-Nibblet 2 (rozhodovacie zoznamy),
NEX – Nonincremental Induction with Exclusion
(rozhodovacie zoznamy),
HCT – Heuristic Criteria Tables (prahové pojmy),
IWP – Iterative Weight Perturbation (prahové pojmy),
SOMA – SelfOrganiser Migrate algorithms,
NCD – Nonincremental Induction of Competitive Disjunctions
(etalóny),
ICD - Incremental Induction of Competitive Disjunctions (etalóny),
NBC – Naive Bayes Classification (pravdepodobnostný pojem),
kNN - k Nearest Neighbours (lenivé učenie – extenzionálna
reprezentácia).

Všetky tieto algoritmy budú charakterizované v podkapitole 2.3 „Najznámejšie algoritmy strojového učenia“. Podrobnejší opis vymenovaných algoritmov je možné nájsť aj v [Machová, 2002]. Na základe podobnosti všeobecného opisu týchto algoritmov boli definované nasledovné základné princípy kontrolovanej klasifikácie: usporiadanie priestoru pojmov, horolezecký princíp, delenie priestoru príkladov na podpriestory, riadenie výnimkami, súťaživý princíp, skórovacia funkcia a redukcia počtu kandidátov. Základné princípy sa medzi sebou môžu kombinovať.

Usporiadanie priestoru pojmov. Jeden zo spôsobov, ako získať hľadanú definíciu pojmu, je vytvoriť priestor všetkých možných definícií pojmov (krátko priestor pojmov), ktoré prichádzajú do úvahy ako riešenia a ten prehľadať. Priestor pojmov sa spravidla zobrazuje ako acyklický graf –

strom. Prehľadávanie môže byť efektívnejšie, ak sa pojmy domény usporiadajú podľa konkrétneho kritéria. Najčastejšie používaným kritériom je všeobecnosť. Pri takomto usporiadaní sa v listoch stromu nachádzajú jednotlivé pozorovania (konkrétne objekty, trénovacie príklady), ktoré sa vyznačujú najnižšou úrovňou všeobecnosti. Kandidáti riešenia, teda kandidáti hľadanej definície pojmu (krátko kandidáti pojmu), sa nachádzajú spravidla v medziľahlých uzloch. Koreň takého stromu obsahuje nultú hypotézu, teda najvšeobecnejší pojem, ktorý pokrýva všetko. Na druhej strane v príznakovom priestore, sú jednotlivé trénovacie príklady reprezentované bodmi a pojmy, alebo definície pojmov predstavujú oblasti, ktoré obsahujú viac konkrétnych príkladov, teda bodov. Prehľadávanie priestoru pojmov usporiadaného podľa všeobecnosti je možné vykonať troma spôsobmi: od všeobecného k špecifickému, od špecifického k všeobecnému a prehľadávanie oboma smermi súčasne. V strojovom učení sa taktiež uplatňujú všetky klasické prehľadávacie metódy umelej inteligencie (prehľadávanie do šírky, hĺbky a pod.). Tento princíp využívajú učiace algoritmy generujúce logické konjunkcie: VSS, EGS a ESG.

Horolezecký princíp. Tento princíp je založený na gradientnom hľadaní extrému v lokálnom okolí aktuálneho bodu. V každej iterácii je preskúmané okolie aktuálneho (najlepšieho doposiaľ nájdeného) riešenia, ktoré je reprezentované bodom v priestore pojmov. Riešenie sa následne presunie do najsľubnejšieho bodu tohto okolia. Sľubnosť riešenia sa môže merať skórovacou funkciou. Výhodou týchto algoritmov je, že dokážu nájsť riešenie mnohých úloh pomerne rýchlo. Ich nevýhodou je, že síce dokážu nájsť lokálny extrém rýchlo, ale často v ňom uviaznu, keď sa v okolí nenachádza lepšie riešenie. Preto sa používajú často na dotiahnutie do extrému, ktorého okolie bolo nájdené inou metódou. Príkladmi algoritmov, využívajúcich tento princíp sú algoritmy: IWP (generuje prahové pojmy), SADE, SOMA [Zelinka, 2002].

Delenie priestoru príkladov na podpriestory. Priestor trénovacích príkladov sa rekurzívne delí na podpriestory až dovtedy, kým nie je splnená ukončovacia podmienka. Napríklad, kým každý podpriestor neobsahuje iba príklady jednej triedy. Delenie na podpriestory sa uskutočňuje väčšinou pomocou informačnej teórie zohľadnením informačného zisku skúmaného delenia. Tento princíp sa využíva v nasledovných algoritmoch: NSC,

AQ11 (generujú produkčné pravidlá), ID3, ID5R, C4.5, MDPL (generujú rozhodovacie stromy) a CN2 (generuje rozhodovacie zoznamy).

Riadenie výnimkami. Podstata princípu riadenia výnimkami spočíva v tom, že trénovacie príklady, ktoré sú chybne klasifikované (výnimky), sú zaradené do novovytvorených tried, resp. pseudo-tried a následne opätovne prebehne proces učenia. To sa opakuje dovtedy, kým nie sú všetky trénovacie príklady správne klasifikované, alebo nastal stav, keď nové iterácie neprinášajú lepšie výsledky. Výnimkami sú riadené také algoritmy ako NCD, ICD, ktoré generujú etalóny a NEX, ktorý generuje rozhodovacie zoznamy.

Súťaživý princíp. Tento princíp strojového učenia je založený na hodnotení kandidátov definície pojmu. Vyberá sa tá definícia, ktorá má najlepšie hodnotenie (napríklad najvyššia pravdepodobnosť, najväčšia podobnosť s typickým reprezentantom atď.). Tento princíp je podstatou Naivného Bayesovho klasifikátora a algoritmov generujúcich etalóny: NCD a ICD.

Skórovacia funkcia. Ak je potreba urýchliť prehľadávanie priestoru pojmov, je možné použiť systém s prehľadávacími preferenciami (search bias), ktorý bude niektorých kandidátov riešenia – definície pojmov - uvažovať skôr ako ostatné. Taký systém musí definovať skórovaciu funkciu, ktorá najsľubnejšieho kandidáta pojmu ohodnotí najvyšším číslom. Princíp skórovacej funkcie využívajú algoritmy HGS, HSG (generujú logické konjunkcie) a HCT (generuje tabuľku kritérií). Taktiež ho používajú algoritmy generujúce rozhodovacie stromy ID3, ID5R a C4.5, pričom ako skórovaciu funkciu používajú informačný zisk. Tento princíp využíva aj algoritmus CN2 (generuje rozhodovacie zoznamy aj neusporiadané množiny pravidiel), ktorého skórovacia funkcia hodnotí signifikanciu opisu.

Redukcia počtu kandidátov. Spravidla pri úplnom prehľadávaní priestoru pojmov je v každej ďalšej iterácii algoritmu viac pojmov ako v predchádzajúcej. Priestor pojmov sa takto môže rýchlo rozširovať. Ak nie je potrebné trvať na požiadavke nájdenia optimálneho riešenia, potom nie je nutné prehľadávať celý priestor pojmov. Prehľadávanie priestoru pojmov je možné redukovať v každej iterácii na určitý počet kandidátov pojmov. Ide vlastne o takzvanú metódu lúčového prehľadávania (beam se-

arch). Výhodné je redukovaný počet kandidátov pojmov zadať formou parametra učiaceho algoritmu. Tento parameter sa zvykne označovať ako BS faktor (Beam Size). Ak už bude prehľadávanie redukované iba na niektoré pojmy, mali by to byť tie najsľubnejšie. Preto je vhodné kandidátov pojmov v každej iterácii ohodnotiť skórovacou funkciou. V súvislosti s princípom redukcie počtu kandidátov sú definované takzvané tvrdé preferencie (hard bias) a mäkké preferencie (soft bias). V prvom prípade sú z prehľadávania tvrdo vopred vylúčení kandidáti pojmu s nízkou hodnotou skórovacej funkcie. Dá sa to vyjadriť aj tak, že sú to kandidáti pojmu, ktorí vypadli v procese redukcie počtu kandidátov. V druhom prípade sú niektorí kandidáti pojmu uvažovaní prednostne, ale žiadny z nich nie je vylúčený z procesu prehľadávania. Z tohto pohľadu prehľadávanie uplatnením tvrdých preferencií predstavuje kombináciu princípov skórovacej funkcie a redukcie počtu kandidátov. Prehľadávanie uplatnením mäkkých preferencií používa iba princíp skórovacej funkcie. Taktiež, horolezecký princíp je možné považovať za špeciálny prípad lúčového prehľadávania pri BS=1. Princíp redukcie počtu kandidátov využívajú algoritmy HGS, HSG (generujú logické konjunkcie) a HCT (generuje tabuľku kritérií).

Návrh nového algoritmu. Na základe doteraz uvedených skutočností je možné systematizovať návrh nového učiaceho algoritmu nasledovným spôsobom. Najprv je analyzovaný problém, ktorý je potrebné riešiť. Potom sú zvolené tie zo základných princípov, ktoré vyhovujú danej učiacej úlohe. Napokon je zostavený všeobecný algoritmus, ktorý by vhodným spôsobom kombinoval zvolené princípy.

Podstata návrhu nového algoritmu spočíva teda vo výbere jeho základných princípov, ktoré by bolo vhodné uplatniť s ohľadom na charakter učiacej úlohy. Ak je potrebné riešiť zložitú úlohu s veľkým rozsahom trénovacích údajov, je možné sa rozhodnúť pre prehľadávanie priestoru pojmov, ale s doplňujúcim využitím princípu skórovacej funkcie a redukcie počtu kandidátov. Táto kombinácia princípov je použiteľná aj pri úlohách, kde hrá svoju úlohu čas a riešenie je potrebné urýchliť, pričom nie je nutné trvať na optimálnom riešení. Ak je potrebné riešiť učiacu úlohu, ktorá je charakteristická zašumenými údajmi, je možné použiť princíp skórovacej funkcie alebo princíp riadenia výnimkami. Niekedy je potrebné potýkať sa s problémom veľkého rozptylu príkladov jednej triedy medzi príklady

ostatných tried, čo znamená, že príklady jednotlivých tried nie sú lineárne separabilné. V takom prípade je možné s úspechom použiť princíp delenia priestoru príkladov na pod priestory resp. „rozdeľuj a panuj“, alebo aj princíp riadenia výnimkami. Ak má učiaci problém vlastnosť lineárnej separability, je možné použiť horolezecký princíp.

Niektoré kombinácie základných princípov sú bežné, ako napríklad kombinácia princípu usporiadania priestoru pojmov s princípom skórovacej funkcie a redukciou počtu kandidátov. Je možné tvoriť aj netradičné kombinácie. Napríklad použitie princípu delenia priestoru príkladov na podpriestory a v rámci jednotlivých podpriestorov aplikovať princíp prehľadávania usporiadaného priestoru pojmov. Podrobnejšie o základných princípoch strojového učenia pojednáva [Machová-Paralič, 2003].

2.2 Výpočtová teória učenia

Výpočtová teória učenia sa zvykne označovať skratkou PAC (Probably learning an Aproximatly Correct hypothesis) [Mitchell, 1997]. PAC si kladie nasledovné otázky: Koľko trénovacích príkladov je potrebných pre úspešné riešenie? Je možné určiť počet chýb pred učením cieľovej funkcie? Je možné určiť výpočtovú zložitosť úloh? Kvôli nájdeniu odpovedí je potrebné zaoberať sa predovšetkým *komplexnosťou príkladov* (počet trénovacích príkladov potrebných na konvergenciu k úspešnej hypotéze), *výpočtovou komplexnosťou* (množstvo výpočtov potrebných na konvergenciu k úspešnej hypotéze) a *ohraničením chybovosti* (prípustný počet chybne klasifikovaných príkladov v procese konvergencie k úspešnému riešeniu).

Problém je možné definovať nasledovne. Po pozorovaní sekvencie trénovacích príkladov z množiny *X* patriacich k cieľového pojmu z množiny *C*, učiaci sa algoritmus *L* musí na výstupe poskytnúť hypotézu *h* z množiny *H*, ktorá je odhadom *C*, kde:

X je množina trénovacích príkladov nad ktorou je definovaná cieľová funkcia,

C je množina cieľových pojmov,

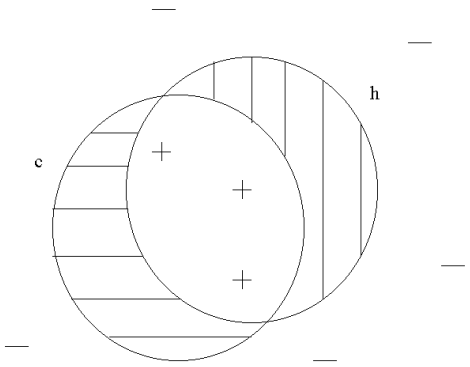
D je distribúcia pravdepodobnosti, na základe ktorej sú vyberané trénovacie príklady do trénovacej množiny,

H je množina možných hypotéz, ktoré učiaci sa algoritmus *L* uvažuje pri pokuse naučiť sa cieľový pojem *C*.

Platí, že pre každé $c \in C$: $x \rightarrow \{0,1\}$. Ak *x* je pozitívny (resp. negatívny) trénovací príklad, potom c*(x)=1(resp. 0)*. *D* nie je známa učiacemu sa algoritmu *L*. Trénovacie príklady sú vyberané podľa *D* a prezentované učiacemu sa algoritmu *L* v tvare *x* spolu s cieľovou funkciou *c(x)*.

Spravidla nie sú k dispozícii všetky trénovacie príklady, ktoré by reprezentovali všetky možné pozorovania a situácie, ktoré v danej doméne môžu nastať. Naviac často sa stáva, že trénovacie príklady sú charakterizované aj pomocou irelevantných atribútov a naopak, nie sú zahrnuté niektoré relevantné atribúty z množiny sledovaných atribútov. To znamená, že naučená hypotéza môže v niektorých prípadoch klasifikovať chybne. Je potrebné odhadnúť chybu naučenej hypotézy. Chybu hypotézy *h „chyba$_D$(h)“* je možné definovať ako pravdepodobnosť toho, že trénovací príklad nebude patriť do oblasti, kde sa naučená hypotéza *h* a cieľový pojem *c* zhodujú v klasifikácii (viď obr.1). Túto chybu je možné vyjadriť nasledovným vzťahom

$$chyba_D(h) = P_{x \in X/D}\left[c(x) \neq h(x)\right] \tag{1}$$



Obr. 1: Definícia chyby hypotézy pojmu v grafickej podobe

Táto definícia platí nad celou distribúciou príkladov, nielen nad trénovacími príkladmi. Takto definovanú chybu je možné určiť ako veľkosť vyšrafovanej plochy v obr.1. V skutočnosti nie je možné exaktne merať veľkosť vyšrafovanej plochy, preto sa chyba hypotézy určuje ako počet príkladov vyšrafovanej plochy, teda počet príkladov nesprávne klasifikovaných, či už ako pozitívne, alebo ako negatívne (*FP* (false positive) + *FN* (false negati-

ve)). Prvkami množiny „*c*“ sú objekty, ktoré tvoria cieľový pojem. Niektoré z týchto objektov sú príkladmi v trénovacej množine, teda vstupno/výstupné vzory *[x,c(x)].*

Chyba hypotézy *h* s ohľadom na *c* nie je priamo pozorovateľná učiacim sa subjektom *L*. Ten môže iba pozorovať prejavy *h* nad trénovacími príkladmi a musí zvoliť výstupnú hypotézu iba na tomto základe.

Chybu, ktorá bola definovaná vyššie, je možné exaktne určiť, ak sú k dispozícii všetky možné pozorovania domény. Ako už bolo spomenuté, tie k dispozícii nie sú. K dispozícii je iba časť trénovacích príkladov v trénovacej množine. Chyba určená z tejto časti trénovacích príkladov je trénovacia chyba. Trénovacia chyba je vzťahovaná k časti trénovacích príkladov chybne klasifikovaných hypotézou *h* na rozdiel od skutočnej chyby definovanej vyššie. Trénovacia chyba súvisí s hodnotením efektívnosti klasifikácie. Pri akýchkoľvek experimentoch súvisiacich s klasifikáciou, resp. zhlukovaním sa očakáva zhodnotenie ich presnosti a návratnosti. Preto sú tieto termíny definované v podkapitole 2.5.

Spoľahlivosť učenia. Úlohou je charakterizovať množinu cieľových pojmov, ktoré môžu byť spoľahlivo naučené a určiť primeraný počet náhodne vybratých príkladov. Aký je teda počet príkladov potrebných na naučenie hypotézy *h* pri chybe nájdenej hypotézy $chyba_D(h)=0$? Opäť je potrebné uvedomiť si, že nie je možné generovať trénovacie príklady zodpovedajúce každému možnému prípadu sledovanej domény. Učiaci sa algoritmus *L* generuje teda viac hypotéz pre rôzne výbery z trénovacej množiny a nevie s istotou určiť jedinú hypotézu odpovedajúcu cieľovému pojmu. Teda existuje nenulová pravdepodobnosť, že trénovacie príklady, s ktorými sa stretne učiaci sa algoritmus *L* budú zavádzajúce. Preto nie je možné očakávať na výstupe nulovú chybu. Ale je reálne požadovať, aby bola chyba ohraničená veľmi malou konštantou ϵ. To znamená, že sa nepožaduje, aby učiaci sa algoritmus *L* uspel pre každú postupnosť ľubovoľne vybratých trénovacích príkladov. Požaduje sa iba, aby pravdepodobnosť jeho zlyhania bola ohraničená malou konštantou δ. Inými slovami, učiaci sa algoritmus *L* vyberie aproximačne korektnú hypotézu. Pre aproximačne korektnú hypotézu *h* je možné formulovať nasledovné tvrdenie.

Cieľový pojem *C* je naučiteľný učiacim sa algoritmom *L* použitím *H*, ak pre každé $c \in C$, pre distribúciu *D* nad *X*, pre ϵ a δ platí, že *L* s pravdepo-

dobnosťou aspoň *(1- δ)* generuje hypotézu pojmu *h* z *H* taký, že pre chybu hypotézy *h* platí $chyba_D(h) \leq \epsilon$.

Komplexnosť trénovacích príkladov v konečnom priestore hypotéz. Nárast požadovaného počtu trénovacích príkladov v závislosti od zväčšovania zložitosti problému sa nazýva „komplexnosť príkladov“ učiaceho problému. V tomto prípade je potrebné určiť hranice komplexnosti príkladov pre triedu *L* nazývanú konzistentný učiaci sa algoritmus. Učiaci sa algoritmus *L* je konzistentný, ak jeho výstupné hypotézy perfektne klasifikujú trénovacie príklady, teda pokrývajú všetky pozitívne príklady cieľového pojmu a nepokrývajú žiaden negatívny príklad cieľového pojmu. Signifikantný priestor pojmov $VS_{H,D}$ obsahuje každú konzistentnú hypotézu v *H* a je definovaný nasledovne $VS_{H,D}=\{h \in H / \forall [x, c(x)] \in X/D: h(x)=c(x)\}$, pričom pravdepodobnosť, že priestor pojmov $VS_{H,D}$ nie je úplný ale je ϵ-úplný je

$$p \leq |H| e^{-\varepsilon m}$$

(2)

kde *m* je počet trénovacích príkladov potrebných na redukciu pravdepodobnosti neúspechu pod požadovanú úroveň δ. Potom platí

$$|H| e^{-\varepsilon m} \leq \delta .$$

(3)

Z uvedenej nerovnosti je možné odvodiť minimálny počet trénovacích príkladov

$$m \geq \frac{1}{\varepsilon}\left(\ln|H| + \ln\frac{1}{\delta} \right)$$

(4)

Ak je uvažovaná hypotéza reprezentovaná konjunkciou binárnych literálov, potom minimálny počet potrebných trénovacích príkladov je možné určiť nasledovne

$$m \geq \frac{1}{\varepsilon}\left(n \bullet \ln 3 + \ln \frac{1}{\delta} \right)$$

(5)

kde $|H|=3^n$ pre n dvojhodnotových literálov. Sú tri možnosti: zahrnúť literál do hypotézy, zahrnúť negáciu literálu do hypotézy a ignorovať literál. Teda je možné konštatovať, že konjunkcie dvojhodnotových literálov sú PAC – naučiteľné.

Ak je uvažovaná hypotéza reprezentovaná DNF, teda disjunkciou konjunkcií *n* binárnych literálov pre maximálny počet *k* konjunkcií, potom početnosť množiny hypotéz je možné odhadnúť nasledovne $|H|=k*3^n$. Počet trénovacích príkladov potrebných pre naučenie hypotéz s pravdepodobnosťou neúspechu pod požadovanou úrovňou δ je

$$m \geq \frac{1}{\varepsilon}\left(n \bullet \ln 3 + \ln k + \ln \frac{1}{\delta} \right).$$

(6)

2.3 Najznámejšie algoritmy strojového učenia

V úvode bolo povedané, že algoritmy strojového učenia je možné deliť podľa toho, či uskutočňujú deduktívne alebo induktívne učenie, respektíve ich kombináciu. Napríklad návrh implementácie kombinujúcej induktívny a deduktívny prístup k strojovému učeniu je uvedený v [Machová, 1995a]. Ďalej bolo povedané, že algoritmy strojového učenia je možné rozdeliť na algoritmy s kontrolovaným učením a nekontrolovaným učením.

Ponúkajú sa aj iné hľadiská delenia učiacich algoritmov, ako delenie na inkrementálne a neinkrementálne. Inkrementálne algoritmy strojového učenia sú schopné spracovávať trénovacie príklady, ktoré prichádzajú postupne, napríklad priamo z konkrétneho procesu. Po príchode každého nového príkladu sa modifikujú doposiaľ známe znalosti (napríklad klasifikačné pravidlá). Neinkrementálne algoritmy spracovávajú všetky trénovacie príklady odrazu. Sú vhodné na riešenie takých úloh, pri ktorých sú všetky trénovacie príklady k dispozícii ešte pred začiatkom učenia.

Ďalší spôsob delenia algoritmov strojového učenia je podľa reprezentácie výstupnej naučenej znalosti. Z tohto hľadiska ich delíme na:

- Algoritmy generujúce **logické konjunkcie**. Výsledná znalosť má formu klasifikačného pravidla, ktorého podmienková časť je konjunkciou podmienok a záverová časť je zaradenie do triedy. K týmto algoritmom patria: algoritmy VSS (Version Space Search) [Mitchell, 1997], ako aj EGS (Exhaustive General to Specific), ESG (Exhaustive Specific to General), HGS (Heuristic General to Specific) a HSG (Heuristic Specific to General) [Langley, 1996].

- Algoritmy generujúce **disjunktívno-normálnu formu**. Výsledná znalosť má formu klasifikačného pravidla, ktorého podmienková časť je disjunkcia konjunkcií podmienok a záverová časť je zaradenie do triedy. Príklady takýchto algoritmov sú: NSC (Nonincremental Separate and Concver [Langley, 1996], AQ11 [Mach, 1997] a AQ15 [Michalski et al., 1986].

- Algoritmy generujúce **rozhodovacie stromy**. Výsledná znalosť má formu rozhodovacej procedúry, ktorá je znázornená acyklickým grafom, teda stromom. Pre klasifikáciu príkladu do triedy, ktorá je zviazaná s niektorým listovým uzlom, musí príklad splniť všetky testovacie podmienky, ktoré sa nachádzajú na ceste z koreňového uzla do daného listového uzla. Najznámejšie algoritmy tejto skupiny sú: ID3 (Iterative Dichotomizer 3) [Quinlan, 1990] a jeho rozšírenie algoritmus C4.5 [Quinlan, 1993]. Tieto algoritmy sú neinkrementálne. Nie veľmi úspešný pokus o inkrementálne generovanie rozhodovacích stromov) [Utgoff-Bradly, 1991] vyústil v algoritmus ID5R (Iterative Dichotomizer 5 Recursive).

- Algoritmy generujúce **rozhodovacie zoznamy**. Výsledná znalosť má formu usporiadaného súboru pravidiel. Každé ďalšie pravidlo sa vkladá do „else“ časti predchádzajúceho pravidla. Preto sa žiadna podmienka nevyšetruje dvakrát. Príklad je zaradený do triedy, ktorá sa nachádza v „then“ časti pravidla, ak sú splnené podmienky „if“ časti pravidla a zároveň nie sú splnené podmienky „if“ častí všetkých predchádzajúcich pravidiel usporiadaného súboru. Príkladmi takých algoritmov sú: NEX (Nonincremental Induction of Decision List with Exclusions) [Langley, 1996] a CN2 [Clark-Nibblet, 1989].

- Algoritmy generujúce **prahové pojmy** definujú prahové kritériá v podmienkovej časti pravidiel, ktoré je nutné splniť, aby príklad mohol byť klasifikovaný do triedy v záverovej časti pravidla. Napríklad tabuľka kritérií definuje *m z n (m ≤ n)* podmienok konjunkcie, ktoré musia byť splnené. Tabuľka kritérií môže byť generovaná algoritmom HCT (Heuristic Criteria Tables) [Langley, 1996]. Lineárna a sférická prahová jednotka určujú hranicu: priamku, respektíve kružnicu či elipsu v dvojrozmernom alebo všeobecnejšie hyperrovinu v n-rozmernom priestore. Táto hranica slúži na diskrimináciu pozitívnych a negatívnych príkladov konkrétnej triedy. Túto hranicu je možné generovať algoritmom IWP (Iterative Weight Perturbation) [Breiman et al., 1984].

- Algoritmy generujúce **etalóny**. Výsledná znalosť má formu typického reprezentanta danej triedy - etalónu. Nový príklad je klasifikovaný do triedy, na etalón ktorej sa najviac podobá, respektíve ku ktorému má v priestore príkladov najbližšie. Etalóny je možné indukovať pomocou algoritmu NCD (Nonincremental Induction of Competitive Disjunction) [Langley, 1996] a ICD (Incremental Induction of Competitive Disjunction) [Bradshaw, 1987].

- Algoritmy generujúce **pravdepodobnostný pojem**. Príklad je klasifikovaný do triedy, ktorej pravdepodobnosť je pre daný príklad najvyššia. Pojem je reprezentovaný pravdepodobnosťou. Nedá sa striktne povedať, že príklad do konkrétnej triedy patrí alebo nepatrí. Príklad patrí teoreticky do každej triedy, len s inou pravdepodobnosťou. Príkladom algoritmu, ktorý generuje pravdepodobnostný pojem je Naivný Bayesov klasifikátor [Mitchell, 1997].

- Algoritmy **lenivého učenia**. Pri tomto prístupe sa negeneruje dopredu žiadna generalizovaná znalosť. Vychádza sa z enumerácie všetkých príkladov rôznych tried. Táto enumerácia reprezentuje znalosti. Keď nastane situácia, že je potrebné rozhodnúť o klasifikácii príkladu, vezmú sa do úvahy niektoré trénovacie príklady, najviac podobné danému príkladu, a ten je zaradený do triedy, do ktorej patrí najviac trénovacích príkladov z jeho blízkeho

okolia. Takéto učenie je realizované veľmi obľúbeným torom kNN (k Nearest Neighbours), ktorý klasifikuje nový príklad na základe k najbližších susedov [Mitchell, 1997].

- Algoritmy **učenia odmenou a trestom** (Reinforcement Learning), niekedy nazývaným aj **posilňovaným učením**, hľadajú cestu v stavovom priestore od počiatočného ku konečnému stavu. Počiatočný a konečný stav je daný. Hľadá sa najkratšia riešiaca cesta, ktorá ich spája. Toto učenie je založené na spätnom šírení odmien, ktoré reprezentujú žiadanosť daného stavu pre hľadané riešenie. Líšia sa spravidla aktualizačnou schémou výpočtu prírastkov odmien. Najznámejšie prístupy sú Q-learning a Bucket Brigade [Kaebling et al., 1996].

- **Zhlukovacie algoritmy**. Trénovacie príklady sa zoskupujú do skupín podľa podobnosti, keďže informácia o zaradení príkladu do triedy nie je známa pred začiatkom učenia. Generované skupiny, teda zhluky môžu neskôr, po ukončení zhlukovania, zastupovať klasifikačné triedy. Príkladmi zhlukovacích algoritmov sú K-means [MacQueen, 1967], [Kanungo, 2002] – najčastejšie používaný algoritmus, ďalej CLUSTER/2 [Michalski-Stepp, 1983] a COBWEB [Fisher, 1987].

Všetky uvedené reprezentácie výsledkov strojového učenia je možné zahrnúť do troch väčších skupín a to:

- **učenie logickej reprezentácie s učiteľom** (logické konjunkcie, disjunktívno - normálna forma, rozhodovacie stromy a rozhodovacie zoznamy),
- **učenie s prvkami kvantitatívneho usudzovania s učiteľom** (prahové pojmy, lineárna a sférická prahová jednotka, etalóny a pravdepodobnostný pojem, lineárna regresia)
- **učenie bez učiteľa** (zhlukovanie a posilňované učenie, resp. učenie odmenou a trestom).

Učenie odmenou a trestom, respektíve posilňované učenie, sa používa na riešenie sekvenčnej úlohy, respektíve úlohy nájdenia minimálnej cesty v grafe. Ale zhlukovanie vedie na riešenie klasifikačnej úlohy. Ide o učenie bez učiteľa, pretože odmena a trest nie sú známe pred začiatkom učenia. Hodnoty odmeny a trestu sa počítajú v procese učenia pomocou aktualizačnej schémy. Podobne všetky ostatné algoritmy z prvých dvoch skupín sa používajú na riešenie klasifikačnej úlohy. Klasifikačná úloha sa zameriava na nájdenie predpisu (napr. klasifikačného pravidla) na klasifikáciu objektu do triedy. Pod triedou si môžeme predstaviť napríklad: oblasť záujmu používateľa internetu, lekársku diagnózu, chybu zariadenia, bonitu klienta, a pod. Sekvenčná úloha, resp. úloha nájdenia minimálnej cesty, predstavuje nájdenie riešenia vo forme najkratšej cesty v stavovom priestore od počiatočného k cieľovému stavu.

Prakticky je možné uvedené algoritmy strojového učenia aplikovať v rôznych oblastiach. Najčastejšie sa používajú v objavovaní znalostí v databázach [Paralič, 2002]. Užitočné znalosti sa dajú takto objaviť pri analýze nákupného košíka, je možné uskutočňovať prieskumy kúpy - schopnosti zákazníkov, ich preferencií pri výbere produktov (tovaru, poistenia, ...) a pod. Algoritmy strojového učenia je možné taktiež aplikovať v každej inej oblasti, kde ide o riešenie klasifikačnej úlohy: spracovanie textových dokumentov, odhaľovanie podozrivých bankových operácií, predikcia vývoja defektov plechov pri valcovaní a tak ďalej.

2.4 Najčastejšie používané algoritmy strojového učenia v spracovaní dokumentov

Táto kapitola sa venuje opisu algoritmov strojového učenia, ktoré sa najčastejšie používajú v systémoch spracovania textových dokumentov. Hlavne však tých, ktoré boli používané pri experimentoch opisovaných vo štvrtej kapitole. Taktiež definícia úlohy a základné pojmy oblasti spracovania textových dokumentov sa nachádzajú v úvode štvrtej kapitoly. V tejto oblasti majú trénovacie príklady formu textových dokumentov.

2.4.1 Klasifikačné algoritmy strojového učenia

Naivný Bayesov klasifikátor [Mitchell, 1997] je založený na teórii pravdepodobnosti. Využíva Bayesovu formulu. Výhodou tohto klasifikátora je použitie jednoduchej pravdepodobnostnej definície triedy. Naopak nevýhodou je predpoklad nezávislosti atribútov opisujúcich klasifikovaný objekt, čo vo väčšine reálnych domén nie je splnené. Aj napriek tejto nevýhode však v doméne klasifikácie textových dokumentov dosahuje naivný Bayesov klasifikátor výsledky kvalitatívne porovnateľné s principiálne zložitejšími algoritmami.

Klasifikátor NBCI (Naive Bayes Classifier using Itemsets). Táto metóda induktívneho strojového učenia bola vyvinutá špeciálne na klasifikáciu textových dokumentov [Kučera-Ježek-Hynek, 2003]. Používa kombináciu Naivného Bayesovho klasifikátora a klasifikátora Itemset. Princíp spočíva vo vyhľadávaní frekventovaných množín termov charakterizujúcich textový dokument, s ktorými sa ďalej pracuje použitím Naivného Bayesovho klasifikátora. Klasifikátor NBCI dosahuje veľmi dobré výsledky pri klasifikácii krátkych textových dokumentov (do 30 významových termov).

kNN klasifikátor „k najbližších susedov“ (k Nearest Neighbours) patrí ku klasifikačným metódam založeným na inštanciách [Mitchell, 1997]. Klasifikátor uchováva v pamäti všetky trénovacie príklady (dokumenty). Klasifikácia prebieha v troch krokoch:
1. V cykle sa vyberie *i*-ty dokument z testovacieho sub-korpusu.

2. Tomuto novému testovaciemu dokumentu sa priradí najpočetnejšia kategória *k* najbližších (v zmysle minimálnej vzdialenosti, resp. maximálnej podobnosti) trénovacích dokumentov. V najjednoduchšom prípade (*1NN*) je teda novému dokumentu priradená kategória najbližšieho dokumentu z trénovacej množiny. V prípade nerozhodnosti (neexistuje práve jedna jednoznačne najpočetnejšia kategória *k* najbližších susedov) sa rekurzívne realizuje *(k-1)NN*, pokiaľ nedôjde k jednoznačnému rozlíšeniu, resp. *k* neklesne na hodnotu *k=1*. Zmienenú nerozhodnosť je možné riešiť aj iným spôsobom (náhodným výberom).
3. Pokiaľ sú klasifikované všetky žiadané príklady, koniec.

Na výpočet vzdialenosti dvoch príkladov sa väčšinou používa kosínusová metrika (viď. koniec tejto podkapitoly). Z algoritmu kNN vyplývajú niektoré skutočnosti:

- výpočtová náročnosť je spôsobená potrebou výpočtu vzdialeností práve klasifikovaného príkladu ku všetkým trénovacím príkladom,
- pamäťová náročnosť je podmienená uchovávaním všetkých trénovacích príkladov v pamäti za účelom následného výpočtu vzdialeností voči testovaciemu príkladu,
- výskyt irelevantných termov, ktoré sa taktiež podieľajú na výpočte vzdialenosti, môže na úkor relevantných negatívne ovplyvniť presnosť klasifikácie.

Na elimináciu týchto aspektov sa používajú rôzne redukčné techniky (CNN, RNN, ENN, IB2, IB3, DROP4 a pod.) a riadené selekcie (redukcie) atribútov pomocou napríklad informačného zisku, vzájomnej informácie, sily termu a ďalších).

2.4.2 Zhlukovacie algoritmy strojového učenia

Zhlukovanie je proces, v ktorom dochádza k zoskupovaniu objektov opísaných pomocou atribútov tej istej množiny do prirodzených skupín (zhlukov) na základe ich vzdialenosti v priestore. Proces sa deje bez apriórnej znalosti tried prislúchajúcich objektom, teda ide o techniku nekontrolovaného učenia. Úlohou je zoskupiť objekty do zhlukov, ktorých počet je buď zadaný pred začiatkom učenia (niektoré algoritmy to vyžadujú), alebo sa

počet zhlukov vykryštalizuje postupne v procese samotného strojového učenia. Teda sám algoritmus vygeneruje najvhodnejší počet zhlukov.

Zhlukovanie je možné rozdeliť na *hierarchické* a *nehierarchické*. Hierarchické zhlukovanie má dva podtypy, *aglomeratívne* a *divízne*. Pri aglomeratívnom zhlukovaní sa vychádza z jednotlivých objektov ako samostatných zhlukov a postupne dochádza k spájaniu zhlukov až po jediný konečný, ktorý obsahuje všetky objekty. Divízne zhlukovanie vychádza z jediného zhluku, ktorý sa v priebehu zhlukovania rozdeľuje až po najnižšiu úroveň, na ktorej každý zhluk obsahuje iba jeden objekt – trénovací príklad.

Nehierarchické zhlukovanie je možné rozdeliť na *optimalizačné* a zhlukovanie pomocou *metódy analýzy modusov*. V prvom prípade sa jedná o hľadanie optimálneho rozkladu vstupnej množiny objektov podľa zvoleného kritéria. Metóda analýzy modusov využíva pravdepodobnostný prístup, pričom zhluky sú definované na základe existencie a polohy modusov frekvenčnej funkcie.

Z hľadiska počítačového spracovania je ešte možné delenie na *paralelné (neinkrementálne)* a *sekvenčné (inkrementálne)* zhlukovacie metódy. Pri paralelných metódach sa pracuje s celou množinou zhlukovaných objektov súčasne, pri sekvenčnom prístupe sa spracovávajú jednotlivé objekty postupne.

Zhlukovací algoritmus k-centier (k-means) je algoritmus založený na myšlienke formovania zhlukov okolo vopred určených príkladov - objektov, ktoré sa v tomto algoritme označujú slovom centrá – „means“ [MacQueen, 1967]. Majme množinu *n* objektov $X = \{x_1, x_2, ..., x_n\}$ a množinu *k* zhlukov $Y = \{y_1, y_2, ..., y_k\}$. Každý objekt je reprezentovaný vektorom v *d*-dimenzionálnom priestore, zhluk je reprezentovaný centrom (etalónom) množiny objektov k nemu patriacich. Centrum zhluku vypočítame podľa vzťahu

$$y_j = \frac{\sum_{x_i \in Y_j} x_i}{|Y_j|} \tag{7}$$

kde $|Y_j|$ je počet objektov patriacich k zhluku Y_j a x_i je *i*-tý objekt z množiny *X*. Je potrebné minimalizovať chybovú funkciu

$$J(X,Y) = \sum_{j=1}^{k} \sum_{i=1}^{n} dist(y_j, x_i) \tag{8}$$

kde funkcia *dist(x, y)* je ľubovoľná metrika. Prehľad najznámejších metrík je uvedený na konci tejto podkapitoly. Algoritmus k-centier (k-means) pozostáva zo štyroch krokov:

1. Inicializácia zhlukov – *k* centier zhlukov sa inicializuje náhodne vybratou množinou *k* objektov.
2. Priradenie objektov – každý objekt sa priradí k najbližšiemu zhluku (v zmysle minimalizácie vzdialenosti , resp. maximalizácie podobnosti objektu a centra zhluku podľa zvolenej metriky).
3. Výpočet nových centier zhlukov – na základe vyššie uvedeného vzťahu sa pomocou aritmetického priemeru vypočítajú nové centrá zhlukov.
4. Ukončovacia podmienka – ak bol dosiahnutý istý počet iterácií, alebo chybová funkcia $J(X,Y) < \varepsilon$ (prah), potom algoritmus končí.

Tento algoritmus predstavuje jednoduchú a obľúbenú zhlukovaciu techniku. Často sa používa pri spracovaní textových dokumentov. Väčšinou dáva veľmi dobré výsledky.

Medzi nevýhody tejto metódy patrí riziko padnutia do lokálneho minima, ktoré je podmienené náhodným inicializačným výberom počiatočných príkladov (napríklad inicializačných dokumentov). K ďalším nevýhodám patrí potreba stanovenia hodnoty parametra *k*, teda počtu zhlukov pred začatím zhlukovania a predpoklad zhluku reprezentovaného sférickým útvarom v *d*-dimenzionálnom príznakovom priestore. Z poslednej nevýhody vyplýva istá citlivosť na zmeny súradníc, čo súvisí s typom použitého váhovania. Miernou modifikáciou (inkrementálna aktualizácia stredov) sa môžu dosiahnuť lepšie výsledky.

Divízne k-centier (Bisecting k-means) rozširuje základné *k*-means a definuje divízne hierarchické zhlukovanie [Kashef-Kamel, 2009]. V každom

kroku sa práve jeden zhluk, vybratý podľa určitého kritéria, rozdelí na dva (bi-sekcia). Proces zhlukovania sa začína s jediným zhlukom, ktorý zahŕňa všetky dokumenty korpusu. Ďalej sa postupuje v štyroch krokoch:

1. Výber zhluku, ktorý sa bude deliť.
2. Nájdenie dvoch sub-zhlukov použitím základného *k*-means algoritmu (tzv. bi-sekčný krok).
3. Opakovanie 2. kroku *n*-krát a pokračovanie v algoritme s rozdelením, ktoré vyprodukovalo zhluky s najväčšou celkovou podobnosťou.
4. Ukončovacia podmienka, opakovanie krokov 1, 2 a 3 pokiaľ sa nedosiahne požadovaný počet zhlukov.

Za zhluk, ktorý sa bude v nasledujúcom kroku deliť, sa vyberá najväčší zhluk, alebo zhluk s najmenšou celkovou podobnosťou, alebo je možné použiť kombináciu oboch kritérií. V niektorých prípadoch je nutné pridať podmienku minimálnej kardinality deleného zhluku, aby nedochádzalo k nerovnomernému zhlukovaniu.

Celkovou podobnosťou zhluku sa rozumie minimálna entropia zhluku alebo maximálna kompatibilita. Entropia na svoj výpočet využíva hodnotu p_{ij}, ktorá vyjadruje pravdepodobnosť, že objekt zhluku *j* patrí do kategórie *i*. Pre zhluk *j* je teda celková entropia definovaná vzťahom

$$E_j = \sum_{\forall i} p_{ij} \log_2(p_{ij}) .$$

(9)

Celková entropia množiny zhlukov sa vypočíta ako suma entropií jednotlivých zhlukov vážená ich kardinalitou

$$E_{celková} = \sum_{j=1}^{k} \frac{n_j E_j}{n} ,$$

(10)

kde *n* je celkový počet objektov, *k* je počet zhlukov a n_j je kardinalita zhluku *j* (počet objektov k nemu priradených).

Výhodami tejto metódy je relatívna jednoduchosť, väčšia presnosť (ako pri predchádzajúcich metódach) a možnosť realizácie rovnako hierarchického ako i nehierarchického zhlukovania.

Samo-organizujúce sa mapy (SOM). Samo-organizujúce sa mapy, tiež známe ako *Kohonenove mapy* [Kohonen, 2001], patria medzi základné modely neurónových sietí používané na nekontrolované učenie. Vizuálne sú tvorené mriežkou (mapou) *n* neurónov, z ktorých každý má priradený vektor váh m_i rovnakého rozmeru ako vstupné vzorky, čím vlastne mapuje *m*-rozmerný priestor do zvyčajne dvojrozmernej štruktúry. SOM realizujú zobrazenie zachovávajúce topológiu. Vstupné dáta sa zobrazujú do mriežky takým spôsobom, že vzdialenosť medzi ľubovoľnými dvoma neurónmi v mriežke zodpovedá Euklidovskej vzdialenosti ich súradníc v priestore. Po natrénovaní dve ľubovoľné blízke vstupné vzorky spôsobujú odozvy na fyzicky blízkych neurónoch výstupnej mriežky.

Proces učenia Kohonenovej mapy pozostáva z adaptácie váh na základe vstupných trénovacích vzoriek. V učiacom cykle sa najskôr vyberajú náhodne vstupné vzorky *x(t)*, ktoré predstavujú vstup a na ich základe sa vypočítajú príslušné aktivácie na všetkých neurónoch. Najčastejšie sa ako aktivačná funkcia používa Euklidova vzdialenosť vektora vstupnej vzorky a váhového vektora práve počítaného neurónu. Neurón s najnižšou hodnotou aktivácie sa označí ako víťaz a práve jeho váhový vektor a váhové vektory susedných neurónov sa budú adaptovať. Adaptácia prebieha na základe vzťahu

$$m_i(t+1) = m_i(t)\alpha(t)h_{ci}(t)\left[x(t) - m_i(t)\right] \tag{11}$$

kde $\alpha(t)$ je koeficient učenia, ktorý s časom klesá a $h_{ci}(t)$ je funkcia susednosti, ktorej parametre sa s časom menia tak, že počet adaptujúcich sa neurónov sa zmenšuje. V procese učenia sa teda váhový vektor približuje k vektoru vstupnej trénovacej vzorky. Praktickým dôsledkom adaptácie nielen váhového vektora víťazného neurónu, ale aj susedných, je už vyššie spomínané topológiu zachovávajúce zobrazenie.

Výhodou klasického SOM prístupu zostáva schopnosť vyjadriť závislosti medzi dokumentmi na základe ich pozície vo výstupnej mape. Nevýhodou SOM máp je, že štruktúra, počet neurónov i dimenzia výstupnej mapy je konštantná a že môžu vzniknúť mŕtve neuróny. SOM sú diskrétne a poskytujú rovnomernú projekciu.

V praxi sa na klasifikáciu dokumentov používajú architektonicky podobné štruktúry a princípy, ktoré na úkor zložitejšieho procesu učenia odstra-

ňujú niektoré nevýhody klasického SOM prístupu. Vznikajú tak GHSOM (Growing Hierarchical Self-Organizing Map), ktoré umožňujú dynamicky rozširovať výstupnú mapu [Dittenbach et al., 2001].

Metriky. Metriky slúžia na výpočet vzdialenosti resp. podobnosti dvoch objektov. Minimalizácia vzdialenosti je ekvivalentná maximalizácii podobnosti. Ak uvažujeme vektorovú reprezentáciu, na výpočet vzdialenosti medzi dvoma vektormi $x=(x_1,x_2,...x_d)$ a $y=(y_1,y_2,...,y_d)$ teda $dist(x,y)$ je možné použiť niektorý z nasledujúcich vzťahov

Manhattanova metrika (cityblock metrika, metrika L1)

$$dist(x,y) = Manh(x,y) = \sum_{i-1}^{d} |x_i - y_i| \tag{12}$$

Euklidova metrika (metrika L2) a druhá mocnina euklidovej metriky

$$sqEucl(x,y) = \sum_{i=1}^{d} (x_i - y_i)^2 \quad Eucl(x,y) = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2} \tag{13}$$

Čebyševova metrika (maximová metrika, L_∞ metrika)

$$L_\infty(x,y) = \max_{i=1}^{d} |x_i - y_i| \tag{14}$$

Minkovského metrika (L metrika)

$$Mink(x,y) = \sqrt[\lambda]{\sum_{i=1}^{d} (x_i - y_i)^\lambda} \tag{15}$$

Kde pre $\lambda = 2$ dostaneme Euklidovu a pre $\lambda = \infty$ Čebyševovu metriku.

Sokalova metrika

$$Sok(x,y) = \sqrt{\frac{1}{d}\sum_{i=1}^{d}(x_i - y_i)^2} \tag{16}$$

Canberra metrika

$$Canbera(x,y) = \sum_{i=1}^{d}\frac{|x_i - y_i|}{|x_i| - |y_i|} \tag{17}$$

Kosínusová metrika podobnosti

$$CosSim(x,y) = \frac{\sum_{i=1}^{d} x_i y_i}{\sqrt{\sum_{i=1}^{d} x_i^2 \sum_{i=1}^{d} y_i^2}} \tag{18}$$

Kosínusová metrika vzdialenosti

$$CosDist(x,y) = e^{-CosSim(x,y)} \tag{19}$$

Metriky, uvedené vyššie, sú použiteľné na výpočet vzdialenosti medzi ľubovoľnými vektormi s numerickými atribútmi. Viac o týchto metrikách je možné nájsť v [Řezanková-Húsek-Snášel, 2007]. V našich experimentoch s kolekciami textových dokumentov bola na výpočet podobnosti používaná výhradne kosínusová metrika.

2.5 Hodnotenie efektívnosti klasifikácie

V rámci tejto práce bude ilustrovaná efektívnosť klasifikácie na príklade klasifikácie textových dokumentov. Avšak túto metodiku je možné použiť na hodnotenie klasifikácie aj v akejkoľvek inej oblasti. Hodnotenie efektívnosti klasifikácie sa môže zakladať na hodnotení stupňa zhody medzi odhadovanou a skutočnou hodnotou funkcie $\Phi(d_i, c_j)$ vypočítanou nad všetkými textovými dokumentmi danej kolekcie dokumentov. $\Phi(d_i, c_j)$ je funk-

cia, ktorej hodnota je *true* (resp. *false*) keď dokument d_i patrí (resp. patrí) do kategórie c_j. Podrobnejšia definícia problému kategorizácie textových dokumentov sa nachádza v úvode 4. kapitoly. Efektívnosť klasifikácie je možné kvantitatívne ohodnotiť pomocou presnosti a návratnosti. Za účelom klasifikácie dokumentov do triedy c_j je možné definovať presnosť π_j ako pravdepodobnosť toho, že platí $\Phi(d_i, c_j)$ podmienenou platnosťou odhadu $\hat{\Phi}(d_i, c_j)$, teda $Pr(\Phi(d_i, c_j) = true \mid \hat{\Phi}(d_i, c_j) = true)$. Podobne návratnosť ρ_j je možné definovať opačnou podmienenou pravdepodobnosťou $Pr(\hat{\Phi}(d_i, c_j) = true \mid \Phi(d_i, c_j) = true)$. Presnosť π_j a návratnosť ρ_j môže byť vyčíslená na základe údajov z kontigenčnej tabuľky (tab.2) nasledovne

$$\pi_j = \frac{TP_j}{TP_j + FP_j}, \tag{20}$$

$$\rho_j = \frac{TP_j}{TP_j + FN_j} \tag{21}$$

kde TP_j *a* TN_j sú počty správne klasifikovaných pozitívnych a negatívnych príkladov triedy (kategórie) c_j. Podobne FP_j *a* FN_j sú počty nesprávne klasifikovaných pozitívnych a negatívnych príkladov kategórie c_j.

Tab.2: Kontigenčná tabuľka pre kategóriu c_j

	$\Phi(d_i, c_j) = true$	$\Phi(d_i, c_j) = false$
$\hat{\Phi}(d_i, c_j) = true$	TP_j	FP_j
$\hat{\Phi}(d_i, c_j) = false$	FN_j	TN_j

Celkovú presnosť a návratnosť pre všetky triedy je možné vyčísliť pomocou mikro-priemeru a makro-priemeru. Mikro-priemer je definovaný nasledovne

$$\pi^{\mu} = \frac{TP}{TP + FP} = \frac{\sum_{j=1}^{|C|} TP_j}{\sum_{j=1}^{|C|} (TP_j + FP_j)}$$
$$\rho^{\mu} = \frac{TP}{TP + FN} = \frac{\sum_{j=1}^{|C|} TP_j}{\sum_{j=1}^{|C|} (TP_j + FN_j)} \quad . \tag{22}$$

Makro-priemer je možné vyčísliť pomocou nasledovných vzťahov

$$\pi^{M} = \frac{\sum_{j=1}^{|C|} \pi_j}{|C|} \qquad \rho^{M} = \frac{\sum_{j=1}^{|C|} \rho_j}{|C|} \quad . \tag{23}$$

Výber konkrétneho spôsobu spriemerovania presnosti a návratnosti závisí od danej úlohy. Napríklad mikro-priemer je vhodný pri práci s frekventovanými triedami, zatiaľ čo makro-priemer je senzitívnejší na klasifikáciu do menej frekventovaných tried. Presnosť a návratnosť je možné kombinovať do miery F_β podľa nasledovného vzťahu

$$F_\beta = \frac{(\beta^2 + 1)\pi\rho}{\beta^2\pi + \rho} \tag{24}$$

kde parameter β vyjadruje pomer vplyvu medzi π and ρ. Často sa používa funkcia F_1, ktorá kombinuje vplyv presnosti a návratnosti s rovnakou váhou.

V prípade, že máme nedostatok trénovacích údajov, teda nie je možné zostaviť dostatočne reprezentatívnu trénovaciu množinu, môžeme odhadnúť efektívnosť klasifikácie použitím krížovej validácie. Pri krížovej validácii je celková trénovacia množina Ω rozdelená na k testovacích podmnožín $T_1, ..., T_k$. Pre každú podmnožinu je generovaný, respektíve naučený klasifikátor Φ_i , ktorý je testovaný na doplnku tejto podmnožiny Ω - T_i . Výsledný odhad efektívnosti (presnosti alebo návratnosti) je vypočítaný ako priemerná hodnota efektívnosti všetkých čiastkových klasifikátorov Φ_i.

Tab.3: Kontigenčná tabuľka pre kategóriu c_j

	expert = áno	*expert = nie*
Klasia. = áno	*A*	*B*
Klasif. = nie	*C*	*D*

Uvažujme kontigenčnú tabuľku v jej čitateľnejšej forme, ktorá nepracuje s pojmom funkcie a jej odhadu, ale s názorom experta a výkonom klasifikátora, ktorý predstavuje odhad názoru experta. Táto tabuľka je ilustrovaná tab.3. A je počet správne pozitívne klasifikovaných príkladov, B je počet nesprávne pozitívne klasifikovaných príkladov, C je počet nesprávne negatívne klasifikovaných príkladov a D je počet správne negatívne klasifikovaných príkladov. Na základe tejto tabuľky je možné presnosť (precision) π charakterizovať ako počet správnych výsledkov pozitívnej klasifikácie do triedy *j* k celkovému počtu výsledkov pozitívnej klasifikácie tak pozitívnych ako aj negatívnych príkladov. Podobne návratnosť (recall) ρ je možné definovať ako počet správnych výsledkov pozitívnej klasifikácie do triedy *j* k celkovému počtu skutočne pozitívnych príkladov

$$\pi_j = \frac{A}{A+B} \,, \tag{25}$$

$$\rho_j = \frac{A}{A+C} \,. \tag{26}$$

Sú známe aj iné miery efektívnosti klasifikácie ako spád chyby F (Fallout), ktorý určuje počet nesprávne klasifikovaných negatívnych príkladov ku celkovému počtu negatívnych príkladov

$$F = \frac{B}{B+D} \,. \tag{27}$$

Ďalšou mierou je odhad chyby ER (Error Rate) teda počet všetkých nesprávnych klasifikácií k celkovému počtu trénovacích príkladov *n*

$$ER = \frac{B + C}{n}.$$

(28)

Efektívnosť klasifikácie je možné merať aj pomocou správnosti AC (Accuracy) teda počtu všetkých správnych klasifikácií k celkovému počtu trénovacích príkladov *n*

$$AC = \frac{A + D}{n}.$$

(29)

Všetky uvedené miery efektívnosti môžu byť nápomocné pri hodnotení efektívnosti klasifikácie. Sú potrebné aj pri meraní zmien v efektívnosti klasifikácie po aplikácii zloženej klasifikácie, ktorej je venovaná nasledujúca kapitola. V konečnom dôsledku tieto miery efektívnosti klasifikácie môžu skvalitniť proces klasifikácie.

3 ZLOŽENÉ KLASIFIKÁTORY

Predchádzajúca kapitola sa končila hodnotením efektívnosti klasifikácie. A práve táto téma úzko súvisí so zloženou klasifikáciou, ktorá zvyšuje efektívnosť klasifikácie slabých klasifikátorov pomocou hlasovania viacerých slabých, alebo tzv. partikulárnych klasifikátorov o konečnej klasifikácii dokumentu do konkrétnej kategórie. Presnosť partikulárnych klasifikátorov môže byť len o málo vyššia ako presnosť náhodnej klasifikácie. Preto tieto klasifikátory označujeme pojmom slabé. Slabé klasifikátory sú tie, pre ktoré je vhodné hľadať techniky zvyšovania efektívnosti, napríklad techniku zloženej klasifikácie.

Efektívnosť klasifikácie vo všeobecnosti závisí od viacerých skutočností. Hlavne od vhodnosti výberu klasifikačného algoritmu pre danú úlohu a od kvality trénovacej množiny. Kvalita trénovacej množiny závisí od toho, či sú sledované hodnoty všetkých relevantných atribútov, či nie sú medzi atribútmi aj irelevantné atribúty, nakoľko presné sú pozorovania – trénovacie príklady, či nie sú zašumené a pod. Za predpokladu, že trénovacia množina nie je dostatočne kvalitná, aby zabezpečila vysokú presnosť klasifikácie, je možné formovať rôzne výbery z trénovacej množiny. Dva najznámejšie prístupy k budovaniu zloženého klasifikátora sú „bagging a boosting“.

Prvý z nich, nazývaný metóda „bagging“ [Breiman, 1994], formuje rozličné výbery z pôvodnej trénovacej množiny a nad každou podmnožinou – výberom natrénuje partikulárny slabý klasifikátor niektorým zvoleným algoritmom strojového učenia. O výsledku klasifikácie nového prípadu sa rozhodne takzvaným „hlasovaním“ všetkých partikulárnych klasifikátorov.

Ďalší prístup je založený na myšlienke použitia váh trénovacích príkladov pri generovaní konkrétneho výberu trénovacej množiny. Všetky výbery trénovacej množiny obsahujú celú trénovaciu množinu, ale v každom výbere majú príklady iné váhy. Táto myšlienka sa dá jednoducho ilustrovať na príklade: ak niektorý trénovací príklad má váhu 2, akoby sa v trénovacej množine nachádzal dvakrát. Keď má váhu 0, akoby sa v trénovacej množine nenachádzal. Samozrejme vo všeobecnosti váha nemusí byť celočíselná. Tento prístup sa nazýva metóda „boosting“ [Schapire-Freund, 1998], [Schapire-Singer, 1999]. Keď sa formuje nový

výber, príkladom, ktoré boli v predchádzajúcej iterácii nesprávne kované sa priradí väčšia váha. Teda väčší dôraz sa kladie na príklady nesprávne klasifikované a na následnú korekciu ďalším partikulárnym klasifikátorom naučeným z ďalšieho výberu trénovacej množiny s novými váhami. Finálna klasifikácia sa tvorí ako hlasovanie partikulárnych klasifikácií, podobne ako pri metóde „bagging“.

Pri samotnom učení zloženého klasifikátora, sa najprv musia naučiť partikulárne klasifikátory, pretože na základe ich hlasovania sa rozhodne o výslednej klasifikácii nového príkladu zloženým klasifikátorom. Vo vykonaných experimentoch v úlohe partikulárnych, resp. základných klasifikátorov v metódach bagging a boosting bol použitý algoritmus generujúci rozhodovacie stromy C4.5 [Quinlan, 1996]. Experimenty s uvedenými metódami optimalizácie výsledkov základných klasifikačných algoritmov boli aplikované na riešenie problému kategorizácie textových dokumentov [Schapire-Singer, 2000].

3.1 Metóda bagging

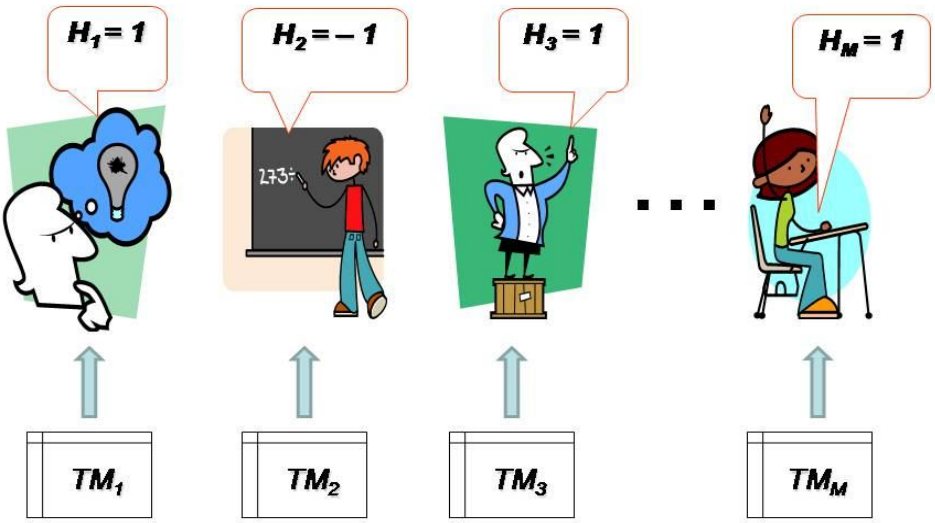
Bagging je metóda na zlepšovanie výsledkov klasifikačných algoritmov, ktorá bola navrhnutá Breimanom [Breiman, 1994]. Jej názov je odvodený zo slovného spojenia “**b**ootstrap **agg**regat**ing**”.

V prípade dvojtriednej klasifikácie, klasifikačný algoritmus generuje klasifikátor *H: D → {-1,1}* na základe trénovacej množiny, teda kolekcie textových dokumentov *D.* Každý dokument je buď klasifikovaný do triedy (1), alebo nie je klasifikovaný do triedy(-1) daným klasifikátorom. Definičný obor klasifikátora *H* je množina dokumentov *D* a obor funkčných hodnôt klasifikátora *H* je množina kategórií *C*.

Metóda bagging vytvára postupnosť *M* partikulárnych klasifikátorov H_m, $m=1,\ldots,M$, kde *M* je počet všetkých výberov (vzoriek) trénovacej množiny. Tieto partikulárne klasifikátory sú kombinované do zloženého klasifikátora nasledovne

$$H(d_i, c_j) = sign\left(\sum_{m=1}^{M} \alpha_m H_m(d_i, c_j)\right). \tag{30}$$

Tento vzťah je možné interpretovať ako hlasovanie partikulárnych klasifikátorov, kde trénovací príklad - dokument d_i je klasifikovaný do triedy c_j, pre ktorú hlasuje väčšina partikulárnych klasifikátorov. Parameter α_m, $m=1,\ldots,M$ posilňuje vplyv presnejších klasifikátorov na konečnú predikciu triedy. Hlasovanie partikulárnych klasifikátorov ilustruje obr.2. Za predpokladu, že počet partikulárnych klasifikátorov je $M=4$, potom podľa obr.2 tri partikulárne klasifikátory hlasujú za zaradenie dokumentu d_i triedy c_j a jeden hlasuje proti ($H_1=H_3=H_4=1$, $H_2=-1$). Teda suma H_m je kladné číslo a výsledok funkcie signum je 1, teda výsledné rozhodnutie zloženého klasifikátora je zaradiť dokument d_i do triedy c_j . Pričom sa predpokladá, že $\alpha_1= \alpha_2= \alpha_3= \alpha_4=1$. Podrobnejšie o hlasovaní klasifikátorov pojednávajú práce [Breiman, 1996] a [Schapire-Freund, 1998].



Obr. 2: Hlasovanie partikulárnych klasifikátorov

Vyššie popísaný postup reprezentuje metódu, ktorá sa nazýva „**základná verzia baggingu**“ [Breiman, 1994]. Avšak, sú známe aj ďalšie podobné stratégie – „**bagging like strategies**“, ktoré rozdeľujú pôvodnú trénovaciu množinu na M rovnako veľkých podmnožín. Nad týmito podmnožinami sa natrénuje M partikulárnych klasifikátorov, ktoré sa združia do jed-

ného zloženého klasifikátora. Tieto stratégie sú ilustrované tab.4, kde originálna trénovacia množina pozostáva zo šestnástich trénovacích príkladov označených veľkými písmenami abecedy.

Tab.4: Baggingové stratégie

Pôvodná množina	**A B C D E F G H I J K L M N O P**			
Disjunktné partície	A B C D	E F G H	I J K L	M N O P
Malé vrecia (MV)	A C H L	B P L P	D I O H	K C F K
MV bez opakovania	A C H L	O P L N	D I O H	K C F P
Disjunktné vrecia	A B C D C	E F G H E	I J K L J	M N O P O

Najznámejšie z týchto stratégií sú disjunktné partície (disjoint partitions), malé vrecia (small bags), malé vrecia bez opakovania (no replication small bags) a disjunktné vrecia (disjoint bags).

Stratégia *disjunktných partícií* používa náhodné rozdelenie príkladov trénovacej množiny do podmnožín, pričom každý príklad sa v podmnožinách nachádza iba raz. Vo všeobecnosti platí, že ak tvoríme *M* podmnožín, každá z nich bude obsahovať *1/M* časť pôvodnej množiny. Zjednotenie podmnožín vytvorí opäť pôvodnú množinu. Klasifikátor *H* získaný zložením partikulárnych klasifikátorov H_m naučených na disjunktných partíciách, dosahuje spravidla najlepšie výsledky zo všetkých baggingových stratégií.

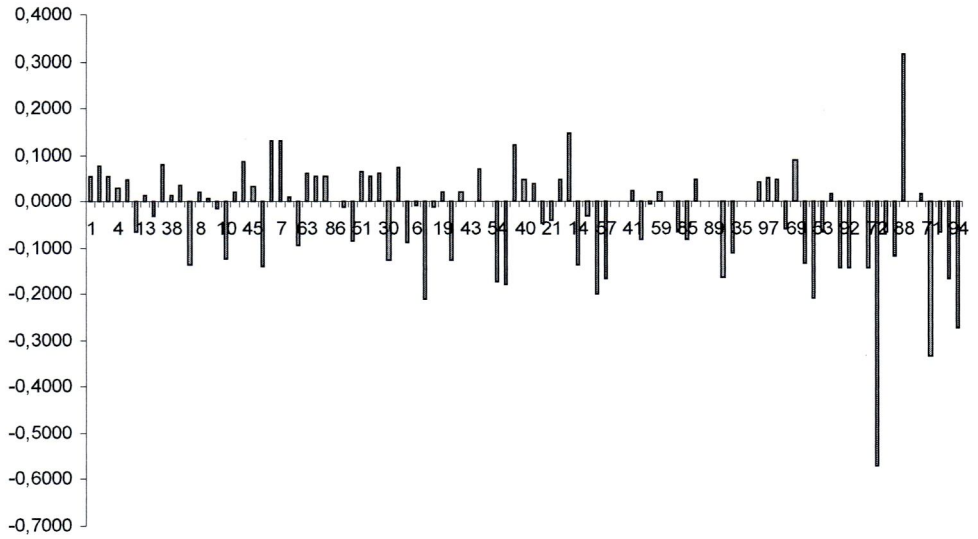
V stratégii *malých vriec* je každá podmnožina príkladov generovaná nezávisle od ostatných náhodným výberom trénovacích príkladov s možnosťou opakovania. Jeden príklad sa môže nachádzať vo viacerých podmnožinách a aj v tej istej podmnožine sa môže vyskytovať opakovane. Zjednotenie partícií nemusí dávať pôvodnú trénovaciu množinu. Táto stratégia dáva najhoršie výsledky.

Stratégia *malých vriec bez opakovania* sa od predchádzajúcej líši tým, že žiadny trénovací príklad sa nenachádza viackrát v tej istej podmnožine.

V poslednej stratégii nazvanej *disjunktné vrecia* sa početnosť podmnožín najprv rovná *1/M*-tej časti pôvodnej množiny. Potom sa ich početnosť zväčší, pretože v každej sa zopakuje náhodne vybratý prvok. Toto zväčšenie nemusí byť iba o jednotku, pretože replikovať sa môže viac prvkov. Počet replikácií musí byť rovnaký v každej podmnožine. Efektívnosť tejto stratégie je podobná efektívnosti disjunktných partícií a za určitých podmienok môže byť aj lepšia.

Bola vykonaná séria testov metódy bagging nad dvoma kolekciami dokumentov: Reuter's-21578 a internetového portálu televízie Markíza. Kolekcia Reuter's-21578 obsahovala 674 kategórií a 24242 termov - slov v anglickom jazyku. Po predspracovaní pomocou lematizácie (steming) a po odstránení neplnovýznamových (stop) slov bol počet termov redukovaný na 19864. Druhá kolekcia – Markíza obsahovala dokumenty klasifikované do 96 kategórií a obsahovala 26785 dokumentov v slovenskom jazyku. Dokumenty oboch kolekcií boli rozdelené na trénovaciu a testovaciu časť v pomere 2:1.

Vykonané testy sa týkali efektívnosti baggingových stratégií a určenia minimálneho počtu základných klasifikátorov potrebných na zlepšenie výsledkov slabej klasifikácie ako aj určenia počtu klasifikátorov, zvyšovaním ktorého sa už efektívnosť klasifikácie výrazne nezvýši.

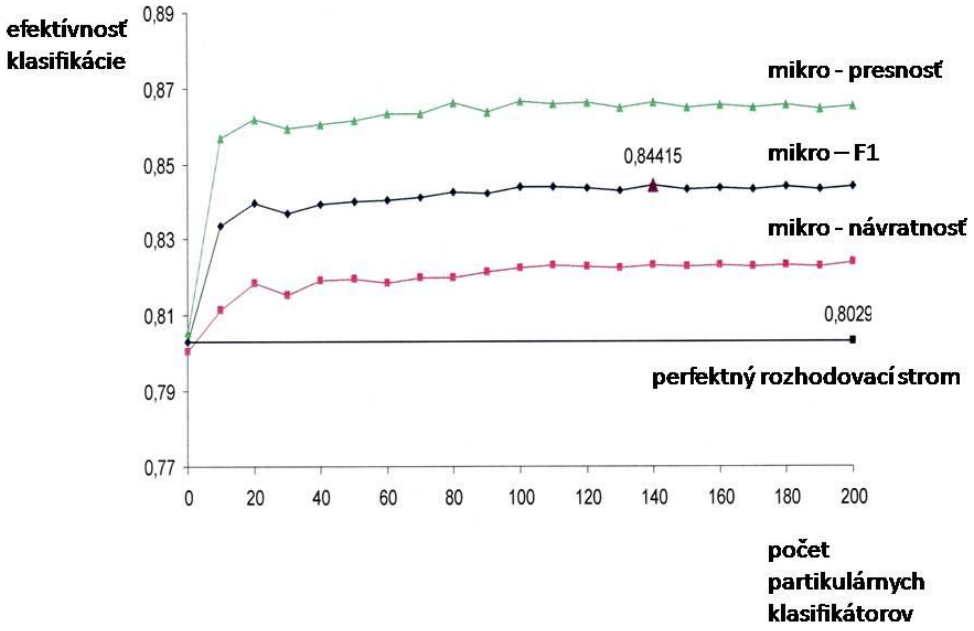


Obr. 3: Rozdiely v presnosti medzi zloženou klasifikáciou založenou na baggingu a perfektným rozhodovacím stromom na kolekcii dokumentov Reuter's

Efektívnosť baggingových stratégií bola overovaná porovnávaním s výsledkami klasifikácie perfektným rozhodovacím stromom. Perfektný rozhodovací strom plnil v týchto experimentoch úlohu silného klasifikátora (neorezaný). Efektívnosti tohto silného klasifikátora sa snažili vyrovnať slabé klasifikátory (orezané) skladaním hlasov väčšieho počtu slabých klasifikátorov v zloženej klasifikácii. Výsledok tohto porovnania je graficky

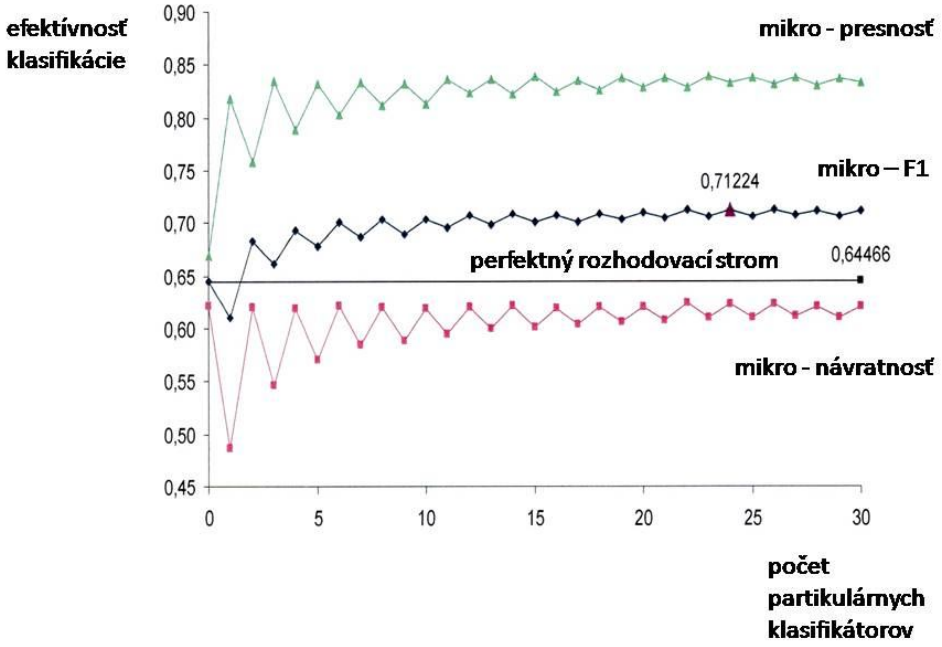
znázornený na obr.3 pre každú triedu zvlášť. Triedy boli usporiadané klesajúco podľa frekvencie ich výskytu zľava do prava. Interpretácia je nasledovná: klasifikácia pomocou baggingových stratégií dáva lepšie výsledky ako perfektný rozhodovací strom pri frekventovanejších kategóriách (ľavá polovica obrázku, kde je viac špičiek smerom nahor). Pri kategóriách s nízkou frekvenciou výskytu dáva lepšie výsledky perfektný rozhodovací strom.

V rámci *určenia minimálneho počtu partikulárnych klasifikátorov* bola skúmaná aj závislosť efektívnosti metódy bagging od počtu klasifikátorov v zloženom klasifikátore. Boli vykonané experimenty pri ktorých bolo najprv uvažovaných 200 klasifikátorov a tento počet bol postupne redukovaný, pričom sa zaznamenávala zmena efektívnosti klasifikácie. Takto získané výsledky (viď obr.4) boli porovnávané s efektívnosťou perfektného rozhodovacieho stromu, ktorá je v obr.4 znázornená rovnou čiarou.



Obr.4: Efektívnosť klasifikácie baggingom v závislosti od počtu partikulárnych klasifikátorov v zloženom klasifikátore na kolekcii dokumentov Reuter's

Z obr.4 je možné vyčítať, že metóda bagging dosahuje stabilné výsledky už pri počte 40 partikulárnych klasifikátorov pre kolekciu Reuter's. To znamená, že zvyšovanie počtu klasifikátorov už nezvýši výrazne efektívnosť klasifikácie. Uspokojivé výsledky by sa dosiahli už pri počte 20 partikulárnych klasifikátorov. Podobné experimenty boli vykonané nad dokumentmi internetového portálu televíznej spoločnosti Markíza. Výsledky sú ilustrované obrázkom obr.5. Z tohto obrázka je možné vyčítať, že 10 až 15 partikulárnych klasifikácií postačí na dosiahnutie stabilných výsledkov zloženej klasifikácie.



Obr.5: Efektívnosť klasifikácie baggingom v závislosti od počtu partikulárnych klasifikátorov v zloženom klasifikátore na kolekcii dokumentov Markíza

Podrobnejší opis experimentov s baggingom je možné nájsť v [Machová et al., 2006] a [Machová-Barčák-Bednár, 2006].

3.2 Metóda boosting

Metódu boosting je možné považovať za zdokonalenie metódy bagging. Podobne ako pri metóde bagging aj metóda boosting generuje postupnosť partikulárnych klasifikátorov H_m, $m=1,\ldots,M$ pre M rôznych výberov trénovacej množiny. Tieto klasifikátory sa kombinujú do výsledného klasifikátora presne tým istým spôsobom ako pri baggingu. Trénovacia množina je modifikovaná prostredníctvom distribúcie váh trénovacích príkladov - dokumentov $d_i \in D$, kde D je kolekcia dokumentov. Množina váh je pred naučením prvého klasifikátora nastavená uniformne. Pre každú iteráciu sa zvýšia váhy tých trénovacích príkladov, ktoré boli nesprávne klasifikované predchádzajúcim klasifikátorom H_{m-1}. Naopak váhy tých príkladov, ktoré boli klasifikované správne sa znížia. Proces učenia sa takto sústredí na nesprávne klasifikované príklady. Najznámejší z boostingových algoritmov AdaBoost.MH2 [Schapire-Singer, 1999] reprezentuje algoritmus klasifikácie do viac ako dvoch tried. Tento algoritmus generuje klasifikátory $H_m: D \times C \rightarrow R$, ktoré predikujú triedy $c_j \in C$ na základe rozhodovacej funkcie $sign[H(d_i, c_j)]$. V experimentoch bol použitý nasledovný boostingový algoritmus:

1. Inicializuj distribúciu váh $w_{1(i,j)} = 1 / (|D||C|)$,
   $i = 1, \ldots, |D|$, $j = 1, \ldots, |C|$
2. Pre $m = 1, \ldots, M$
   2.1. Generuj klasifikátor $H_m: D \times C \rightarrow R$
   2.2. Urči parameter $\alpha_m \in R$.
   2.3. Modifikuj distribúciu váh podľa pravidla

$$w_{m+1(i,j)} = \frac{w_{m(i,j)} \exp(-\alpha_m y_{i,j} H_m(d_i, c_j))}{Z_m} \tag{31}$$

kde Z_m je normovacia konštanta, ktorá garantuje, že platí

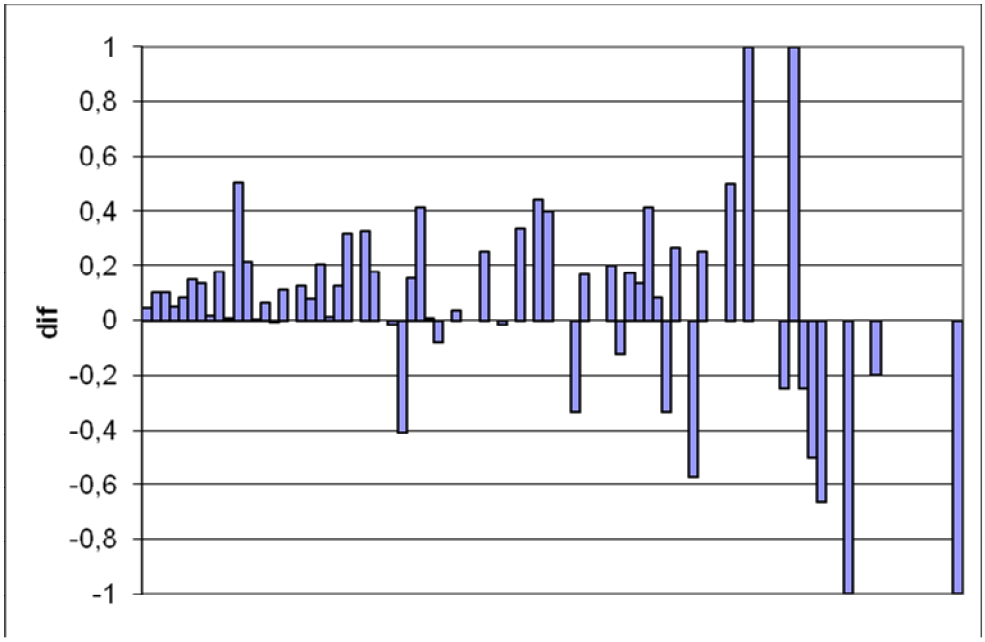
$$\sum_{i=1}^{|D|} \sum_{j=1}^{|C|} w_{m+1(i,j)} = 1 . \tag{32}$$

3. Výstupom je zložený klasifikátor, ktorého matematický model je uvedený vo vzťahu (30).

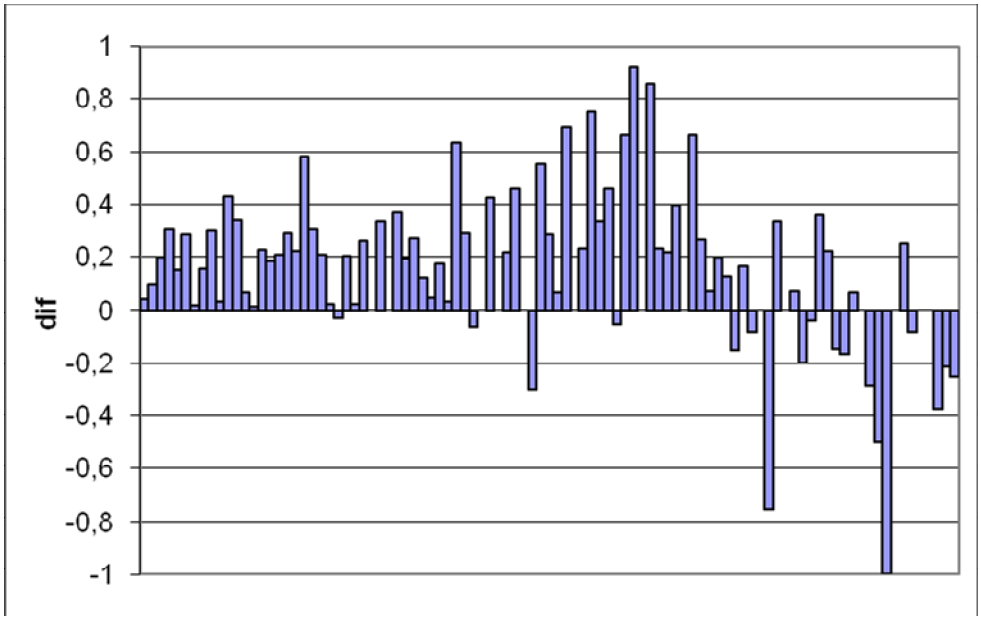
Premenná $y_{i,j}$ je určená ako $y_{i,j} = +1$ ak $d_i \in c_j$ a ako $y_{i,j} = -1$ ak $d_i \notin c_j$. Opísaný algoritmus bol modifikovaný tak, aby výpočet váh nebol ovplyvňovaný nepresnosťou spôsobenou zaokrúhľovaním váh.

Bola vykonaná séria testov metódy boosting nad dvoma kolekciami dokumentov Reuter's-21578 a internetového portálu televízie Markíza. Vykonané testy ukázali zmysel použitia metódy boosting. Pomocou experimentov sa podarilo zistiť minimálny počet partikulárnych klasifikátorov potrebných na zlepšenie výsledkov slabej klasifikácie ako aj počet klasifikátorov, zvyšovaním ktorého sa už presnosť klasifikácie výrazne nezvýši.

V rámci testovania *efektívnosti metódy boosting* bolo zistené, že metóda boosting (kladné hodnoty rozdielu „dif") dáva vo všeobecnosti lepšie výsledky ako perfektný rozhodovací strom (záporné hodnoty rozdielu „dif"), ako ilustruje obr.6 pre kolekciu Reuter's a obr.7 pre kolekciu Markíza.



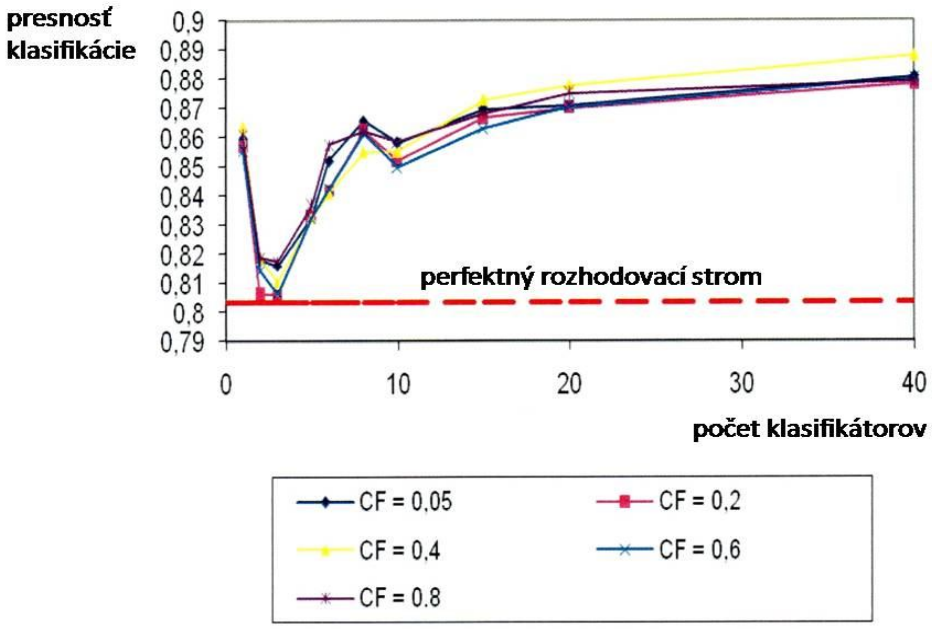
Obr.6: Rozdiely v presnosti medzi zloženým klasifikátorom založenými na boostingu a perfektným rozhodovacím stromom na kolekcii Reuter's



Obr.7: Rozdiely v presnosti medzi zloženým klasifikátorom založeným na boostingu a perfektným rozhodovacím stromom na kolekcii „Markíza“

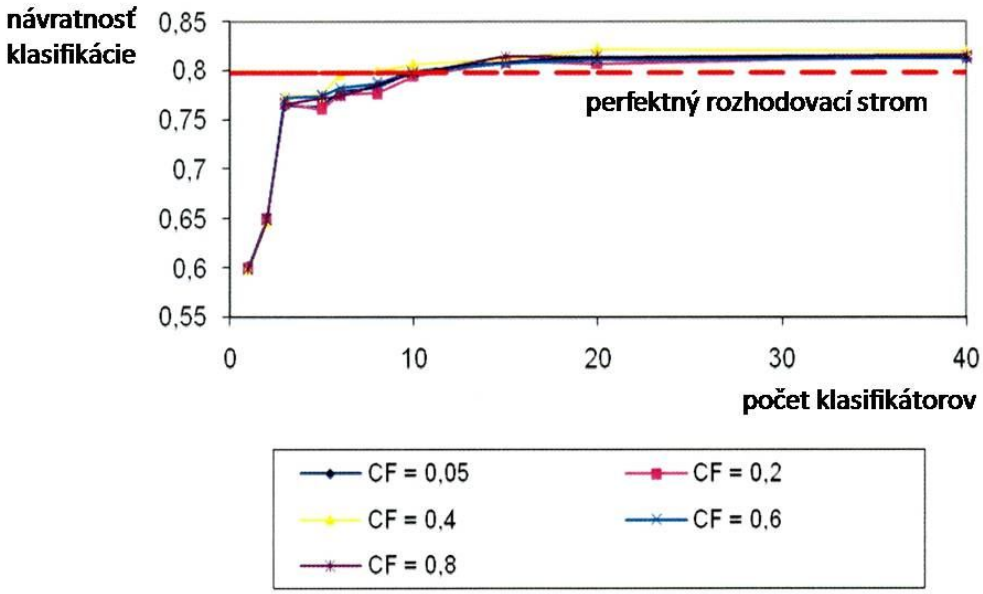
Presnosť klasifikácie bola vyhodnotená pre každú triedu – kategóriu zvlášť. Kategórie boli usporiadané klesajúco podľa frekvencie ich výskytu. Z výsledkov ilustrovaných v obr.6. a obr.7 vyplýva, že zložený klasifikátor (boosting) dáva lepšie výsledky ako perfektný rozhodovací strom pre vyššie frekvencie výskytu. Tento záver je odôvodniteľný, keďže nízky počet výskytov niektorej kategórie v trénovacej množine neposkytuje dostatok informácie pre mechanizmus zloženej klasifikácie.

Bolo potrebné taktiež určiť *minimálny počet partikulárnych klasifikátorov.* Počet partikulárnych klasifikátorov bol limitovaný počtom 100 klasifikátorov. Následne bol počet klasifikátorov znižovaný a bol sledovaný dopad na efektívnosť klasifikácie.



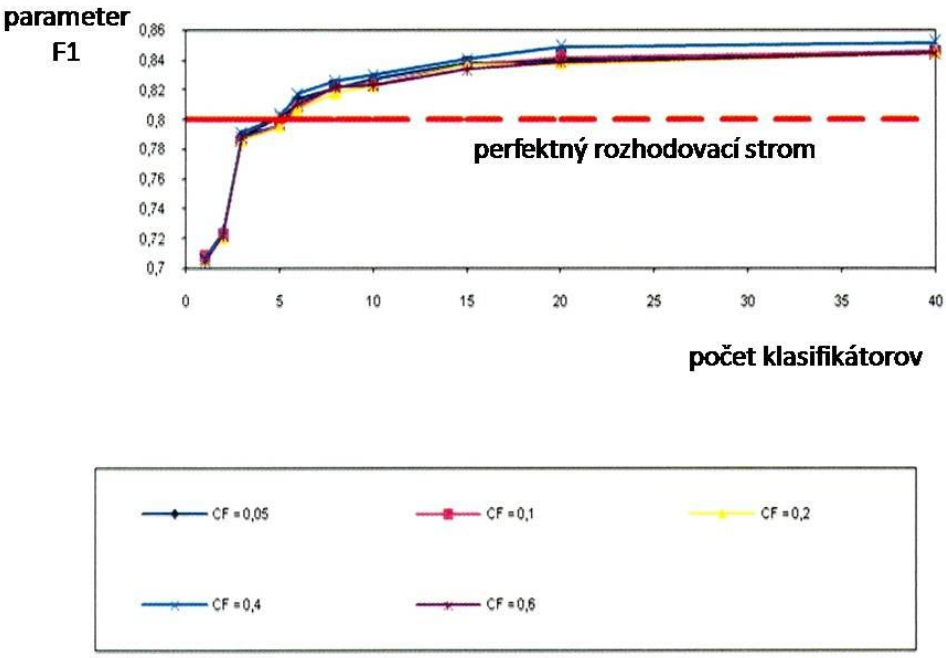
Obr.8: Závislosť presnosti zloženej klasifikácie od počtu základných klasifikátorov v boostingu na kolekcii Reuter's-21578

Obr.10 ilustruje skutočnosť, že už pri približne 5 základných klasifikátoroch dával boosting výsledky porovnateľné s perfektným rozhodovacím stromom (v obr.8, obr.9 a obr.10 – prerušovaná čiara) ak bola efektívnosť zloženej klasifikácie meraná parametrom F_1. Nad počtom približne dvadsať základných klasifikátorov bola efektívnosť zloženej klasifikácie prakticky konštantná a o 5% lepšia ako pri perfektnom rozhodovacom strome. V ďalších parametroch efektívnosti boli dosiahnuté podobné výsledky. Konkrétne, na dosiahnutie uspokojivej presnosti klasifikácie (obr.8) bol potrebný minimálny počet cca 15 klasifikátorov a na dosiahnutie uspokojivej návratnosti klasifikácie (obr.9) bol potrebný minimálny počet cca 10 klasifikátorov.



Obr.9: Závislosť návratnosti zloženej klasifikácie od počtu základných klasifikátorov v boostingu na kolekcii Reuter's-21578

Zaujímavé je, že na obr.8 spočiatku klesá presnosť klasifikácie s rastom počtu klasifikátorov v metóde boosting. Dôvodom je preučenie partikulárneho klasifikátora. Toto preučenie spôsobuje prechodný nárast zložitosti klasifikátora na úroveň perfektného rozhodovacieho stromu. Postupne sa presnosť boostingu vyrovná perfektnému rozhodovaciemu stromu a presiahne ho. Podrobný opis výsledkov uvedených experimentov s boostingom je možné nájsť v [Machová-Puszta-Bednár, 2006].



Obr.10: Závislosť parametra efektívnosti zloženej klasifikácie F1 od počtu základných klasifikátorov v boostingu na kolekcii Reuter's-21578

3.3 Porovnanie metód bagging a boosting

Je potrebné spomenúť, že použitie tak baggingových ako aj boostingových stratégií má zmysel iba vtedy, keď sa pracuje so slabými klasifikátormi. Silný klasifikátor sa nepotrebuje „radiť" s množstvom ďalších klasifikátorov.

Ukázalo sa, že metódy zloženej klasifikácie (bagging a boosting) naozaj môžu zlepšiť výsledky klasifikácie slabými klasifikátormi. Avšak, použitie baggingu a boostingu má aj svoje nevýhody, a to je strata jednoduchosti, prehľadnosti a ilustratívnosti singulárneho rozhodovacieho stromu. Taktiež sa zvyšuje výpočtová zložitosť.

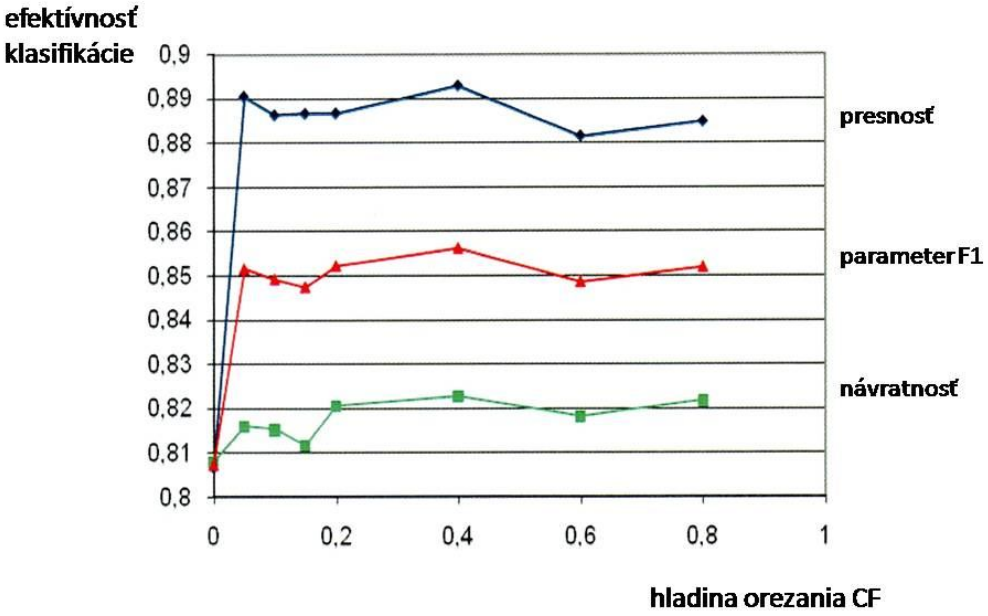
Porovnanie týchto metód preferuje boosting pred baggingom. Bagging je jednoduchšou metódou, avšak boosting dáva lepšie výsledky. Na kolekcii dokumentov Reuter's-21578 sa ukázalo, že uspokojivá efektívnosť

klasifikácie sa dosahuje pri baggingu medzi 20 až 40 základnými fikátormi, zatiaľ čo pri boostingu sa uspokojivá efektívnosť klasifikácie dosahuje medzi 5 až 15 základnými klasifikátormi. Lepšie výsledky boostingu je možné odôvodniť faktom, že táto metóda používa váhovanie trénovacích príkladov. Keďže sa zvyšujú váhy príkladov nesprávne klasifikovaných v predchádzajúcej iterácii, proces učenia sa sústreďuje na nesprávne klasifikované príklady. Tým sa učenie zrýchľuje a spresňuje. Porovnanie oboch metód je uvedené aj v [Machová et al., 2006].

3.4 Orezávanie stromov

V predchádzajúcich podkapitolách boli popísané experimenty s metódami bagging a boosting, teda metódami zloženej klasifikácie, ktoré v úlohe partikulárnych klasifikátorov používali binárne rozhodovacie stromy vygenerované algoritmom C4.5. Avšak tento algoritmus generuje perfektné rozhodovacie stromy. Pre overenie efektívnosti, resp. výkonnosti zloženej klasifikácie bolo potrebné použiť slabé klasifikátory, ako už bolo spomenuté. Slabé klasifikátory boli v experimentoch získané orezaním perfektných binárnych rozhodovacích stromov. Efektívnosť týchto slabých klasifikátorov bola v experimentoch porovnávaná s efektívnosťou perfektného rozhodovacieho stromu, ktorý nebol orezaný. Taktiež bolo sledované či a ako sa zvýši efektívnosť slabých klasifikátorov so zvyšovaním ich počtu v zloženej klasifikácii (bagging a boosting) stále v porovnaní s perfektným rozhodovacím stromom.

V súvislosti s generovaním slabých klasifikátorov vyvstala otázka, ako veľmi ich orezať, teda na akej hladine významnosti *CF* a akým algoritmom. Hladina významnosti v orezávacej technike sa niekedy stručne označuje ako hladina orezania. V našich experimentoch bola použitá konkrétne orezávacia technika „*algoritmus pesimistického odhadu chyby*“, ktorý tento odhad určuje vypočítaním štandardnej odchýlky odhadnutej presnosti za predpokladu binominálneho rozdelenia. Pre danú hladinu významnosti je spodná hranica odhadu vzatá ako veľkosť sily pravidla.



Obr.11: Parametre efektívnosti klasifikácie rozhodovacími stromami v závislosti od hladiny orezania *CF*

Pre veľké množiny dát je pesimistický odhad veľmi blízko k presnosti na trénovacích dátach (štandardná odchýlka je veľmi malá). Napriek tomu, že táto metóda nie je štatisticky úplne korektná, našla si využitie v praxi. Obr.11 ilustruje ako sa mení výkonnosť klasifikátora, konkrétne rozhodovacieho stromu, v závislosti od hladiny orezania *CF*. Najlepšie výsledky boli dosiahnuté pre *CF=0.4*. Preto v rámci experimentov s baggingom a boostingom bola použitá práve táto hladina orezania.

4 STROJOVÉ UČENIE V SYSTÉMOCH SPRACOVANIA TEXTOVÝCH DOKUMENTOV

Táto kapitola sa sústreďuje na spracovanie textových dokumentov z webových zdrojov preto tvorí jadro tejto práce Keď sa otvára problém spracovania textových dokumentov v súvislosti so strojovým učením, väčšinou ide kategorizáciu textových dokumentov. Cieľom kategorizácie textov je nájsť aproximáciu neznámej funkcie $\Phi : D \times C \rightarrow \{true,false\}$, kde D je množina dokumentov a $C = \{c_1, ..., c_{|C|}\}$ je množina preddefinovaných kategórií. Hodnota funkcie Φ sa pre pár $\langle d_i,c_j \rangle$ rovná hodnote *true (resp.false)*, ak dokument d_i patrí (resp. nepatrí) do kategórie c_j.. V skutočnosti sa klasifikačný algoritmus z trénovacích príkladov - dokumentov nenaučí funkciu Φ, ale aproximáciu tejto funkcie $\hat{\Phi}$, ktorú nazývame klasifikátorom.

Definícia kategorizácie textov je založená na nasledovných obmedzeniach:

- Kategórie sú iba nominálne nálepky a neexistuje žiadna vedomosť (či už deklaratívna alebo procedurálna) o ich význame.
- Kategorizácia je založená na extrakcii znalostí z textu dokumentov.

V závislosti od konkrétnej aplikácie je možné limitovať počet kategórií, pre ktoré má funkcia Φ hodnotu „true“ pre daný dokument d_i. Ak dokument d_i môže byť klasifikovaný do práve jednej triedy $c_j \in C$, potom C reprezentuje množinu disjunktných tried. Ak každý dokument môže byť klasifikovaný do $k = 0, ..., |C|$ tried z množiny C, potom ide o multitriednu klasifikáciu a C reprezentuje množinu prekrývajúcich sa tried.

Binárna klasifikácia reprezentuje špeciálny prípad multitriednej klasifikácie a to klasifikáciu do dvoch tried. V princípe môže byť algoritmus binárnej klasifikácie použitý aj pre multitriednu klasifikáciu. Ak sú triedy nezávislé od seba (t.j. pre každú dvojicu tried c_j, c_k a $j \neq k$ je hodnota $\Phi(d_i, c_j)$ nezávislá od hodnoty $\Phi(d_i, c_k)$), problém multitriednej klasifikácie je možné dekomponovať na $|C|$ nezávislých binárnych klasifikácií do tried $\{c_i, \overline{c}_i\}$ pre $i = 0, ..., |C|$.

4.1 Reprezentácia textových dokumentov

Pre úspešné spracovanie textových dokumentov je potrebná vhodná voľba reprezentácie dokumentov. Klasický prístup definuje dokument množinou kľúčových slov – termov. Potom je možné reálny dokument opísať pomocou vektora váh, ktoré vyjadrujú dôležitosť každého termu množiny kľúčových slov pre definíciu daného dokumentu. Tento vektor nazývame aj lexikálnym profilom dokumentu. Teda ľubovoľný dokument je reprezentovaný vektorom (lexikálny profil) $\vec{d}_i = (w_{i1}, w_{i2}, ..., w_{ij}, ..., w_{in})$ v *n*-rozmernom priestore R^n, kde w_{ij} je váha vyjadrujúca dôležitosť *j*-tého termu v *i*-tom dokumente, ktorej hodnota je nenulová v prípade výskytu *j*-tého termu v *i*-tom dokumente. Medzi základné modely reprezentácie textového dokumentu patrí: booleovský, pravdepodobnostný a vektorový model [Chakrabarti, 2003].

Booleovský model redukuje všeobecný model reprezentácie len na použitie binárnych hodnôt vektora váh. Prvky vektora váh, ktoré definujú dokument, môžu nadobúdať len hodnoty "*1*" alebo "*0*". Model teda berie do úvahy len výskyt, resp. absenciu daného termu v dokumente. Výhodou je jednoduchosť, avšak naproti tomu dosť relevantnou nevýhodou je to, že sa neuvažujú informácie o frekvencii výskytu daného termu v dokumente. Tento problém narastá pri veľkých dokumentoch, kde je aj počet termov veľký. V booleovskom modeli sú nevýznamné termy rovnako dôležité ako významné - charakteristické termy dokumentu.

Pravdepodobnostný model. V situácii, keď je k dispozícii korpus dokumentov, v ktorom je potrebné vyhľadať konkrétny dokument, je vhodné použiť práve pravdepodobnostný model. Tento model dobre poslúži úlohe zoradenia nájdených dokumentov podľa pravdepodobnosti relevancie kľúčového slova k danému dokumentu, teda dotazu na daný dokument. Dokumenty korpusu, resp. dokument - dotaz sú reprezentované binárnymi vektormi $\vec{d}_j$ resp. *q*. Podobnosť je definovaná

$$sim(d_j, q) = \frac{P(R \mid d_j)}{P(\overline{R} \mid d_j)} \tag{33}$$

kde $P(R|d_j)$ je pravdepodobnosť, že dokument d_j je relevantný k dotazu. Ďalej *R* je množina relevantných a $\overline{R}$ množina nerelevantných dokumentov vzhľadom k dotazu. Použitím Bayesovej teorémy za predpokladu lineárnej nezávislosti atribútov je možné odvodiť

$$sim(d_j.q) = \frac{P(R)}{P(\overline{R})} \prod_{i=1}^{n} \frac{P(t_i \mid R)}{P(t_i \mid \overline{R})}$$

(34)

kde t_i je *i*-tý term z množiny všetkých *n* termov použitých v modeli. $P(t_i|R)$ je pravdepodobnosť výskytu termu t_i v *R* pre dotaz *q* a *P(R)* je pravdepodobnosť relevancie dokumentu (pre každý dokument je to konštanta). Ďalšími úpravami je možné odvodiť

$$sim(d_j, q) = \sum_{i=1}^{n} \log \frac{P(t_i \mid R)P(\bar{t}_i \mid \overline{R})}{P(t_i \mid \overline{R})P(\bar{t}_i \mid R)}.$$

(35)

Keďže množina *R* na začiatku procesu nie je známa, je nutná nasledovná inicializácia

$$P(t_i \mid R) = 0.5$$

(36)

$$P(t_i \mid \overline{R}) = \frac{n_i}{N}$$

(37)

kde n_i je počet dokumentov obsahujúcich term t_i a *N* je celkový počet dokumentov v korpuse. V tomto kroku model vráti počiatočnú množinu relevantných dokumentov, ktorá sa ďalej spresňuje. Nech *V* je množina dokumentov vrátených v iterácii *(i – 1)* a V_i je množina vrátených dokumentov obsahujúcich term t_i. Potom inicializácia v kroku *i* bude nasledovná

$$P(t_i \mid R) = \frac{|V_i|}{|V|}$$

(38)

$$P(t_i \mid \overline{R}) = \frac{n_i - |V_i|}{N - |V|} .$$

(39)

Nevýhodou tohto modelu je predpoklad nezávislosti atribútov a binárna reprezentácia vektorov dokumentov i dotazu. Podrobnejšie informácie o tomto modeli je možné nájsť v [Han et al., 2005].

Vektorový model. Vektorový model (Vector Space Model / VSM) [Salton et al., 1975] je najčastejšie používanou reprezentáciou textových dokumentov. Príznakový priestor pre túto reprezentáciu je konštruovaný na základe množiny slov, pričom každý príznak korešponduje s textovou jednotkou v dokumente. Za textovú jednotku je považované slovo, fráza, resp. iná lexikálna jednotka. Daný model vychádza len zo štatistických charakteristík dokumentu, a to najmä z distribúcie pravdepodobnosti jednotlivých slov extrahovaných z dokumentov. Reprezentácia dokumentu predstavuje n - rozmerný vektor $d_i = (w_{i1}, w_{i2}, \ldots, w_{in})$ pričom w_{ij} je funkčná hodnota vyjadrujúca váhu, resp. dôležitosť *j* - tého slova v *i* – tom dokumente d_i a *n* je rozmer indexovej množiny slov, t.j. tých slov, ktoré tvoria príznaky opisu jednotlivých dokumentov. VSM odstraňuje nedostatky booleovského modelu v tom, že jednotlivé váhy reflektujú nielen výskyt príslušného termu (booleovsky *„0“, „1“*), ale opisujú aj počet výskytov príslušného termu v dokumente, resp. jeho váhu, teda jeho dôležitosť či významnosť. Vektorový model dokumentu neberie do úvahy štruktúru dokumentu, kontextovú a ani sémantickú informáciu o danom terme, keďže vychádza iba zo štatistických charakteristík dokumentu.

Množinu *m* dokumentov (korpus) môžeme opísať maticou *A* o rozmere *m* x *n*

$$A = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_i \\ \vdots \\ d_m \end{pmatrix} = \begin{pmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,j} & \cdots & w_{1,n} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,j} & \cdots & w_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ w_{i,1} & w_{i,2} & \cdots & w_{i,j} & \cdots & w_{i,n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ w_{m,1} & w_{m,2} & \cdots & w_{m,j} & \cdots & w_{m,n} \end{pmatrix}. \quad (40)$$

Nech platí, že $F = A^T$. Potom pre váhu w_{ij} termu t_j v dokumente d_i platí $w_{ij}=F(d_i, t_j)$. Funkciu *F* nazývame váhovou funkciou. Konkrétna definícia tejto funkcie reprezentuje rôzne spôsoby váhovania termov. Podľa [Dumais, 1991] sa váhovanie rozkladá na lokálne váhovanie, teda váhovanie na základe počtu výskytov termov v samotnom dokumente $L(d_i,t_j)$ a na globálne váhovanie $G(t_j)$ vyjadrujúce počet výskytov termu v celom korpuse. Váhovaná frekvencia termu t_j v dokumente d_i je súčinom lokálnej váhy a globálnej váhy $w_{ij}=L(d_i,t_j)G(t_j)$.

V experimentoch uvádzaných v rámci tejto práce bol použitý vektorový model reprezentácie textových dokumentov.

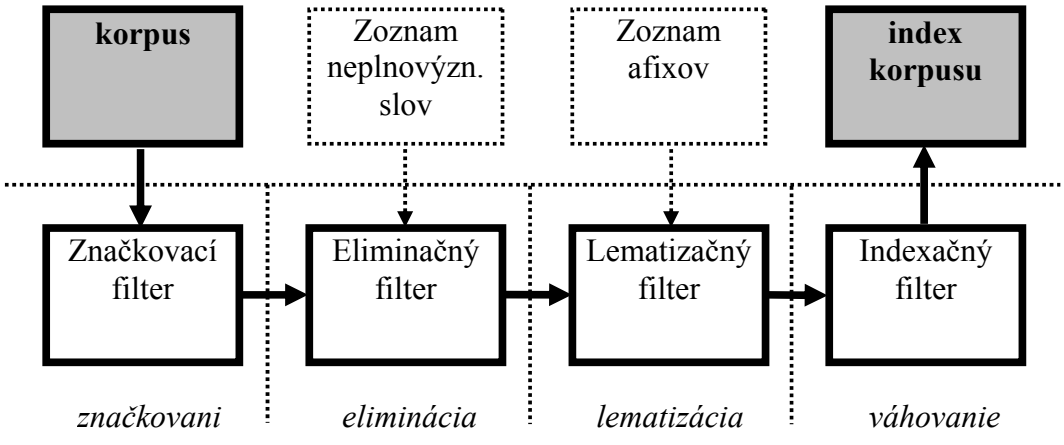
4.2 Predspracovanie textových dokumentov

Spracovaniu textových dokumentov musí predchádzať predspracovanie textových dokumentov. Predspracovanie je proces, v ktorom je dokument efektívne transformovaný do vhodnej formy podľa zvoleného typu reprezentácie (napríklad do formy indexovaného dokumentu). Indexovaný dokument môže byť vytvorený *manuálnou indexáciou* alebo *automatickou indexáciou*. Pri manuálnej indexácii ide o manuálny úkon, za ktorým je rozhodnutie, vyžadujúce špecifické vedomosti. Manuálna indexácia je veľmi zložitý proces, ktorý závisí od množstva subjektívnych faktorov. Preto je potrebná aj automatická indexácia. V súčasnosti automatická indexácia nemôže nahradiť manuálnu indexáciu ale môže predstavovať určitý druh podpory manuálnej indexácie. Automatická indexácia sa delí na dva typy: automatickú extrakciu (indexáciu slov - termov na základe štatistického prístupu po extrakcii indexačných termov priamo z dokumentu)

a automatické priradzovanie (indexáciu pojmov na lingvistickom prístupe, pričom indexačné termy sa extrahujú z riadeného slovníka po jeho porovnaní s textom dokumentu).

V našich experimentoch bola použitá automatická extrakcia, ktorá pozostávala z niekoľkých krokov: lexikálnej analýzy (značkovania, resp. tokenizácie), eliminácie neplnovýznamových (stop) slov, lematizácie (stemming) a váhovania. Lexikálna analýza bola pre potreby našich experimentov vykonaná pomocou "Lower case filter", eliminácia neplnovýznamových, nešpecifických slov, bola uskutočnená prostredníctvom "Stop words filter", lemmatizácia bola vykonaná prostredníctvom "Stem filter" a napokon váhovanie bolo uskutočnené pomocou "Index filter". Všetky tieto filtre sú súčasťou knižnice "Jbowl". Knižnica "Jbowl" je originálny softvérový balík vyvinutý v Jave na podporu úlohy spracovania textových dokumentov. Táto knižnica je podrobnejšie popísaná v [Bednár et al., 2005]. Schéma procesu predspracovania textových dokumentov je znázornená na obr.12.

Lexikálna analýza. V tomto procese sa v súvislom texte dokumentu rozpoznávajú jednotlivé slová, resp. celé spojenia slov - frázy, ktoré majú väčšiu informačnú hodnotu. Najčastejšie sú slová oddeľované pomocou medzier, ale existujú texty, v ktorých takéto rozpoznávanie vedie k nejednoznačným výsledkom.



Obr.12: Schéma procesu predspracovania textových dokumentov

Na identifikáciu spojení slov sa používa niekoľko metód, napríklad: štatistická identifikácia – slová, ktoré sa v texte dokumentu i v celom korpuse vyskytujú často spolu sa považujú za frázy. Sleduje sa *frekvencia výskytu frázy* (záleží na poradí jednotlivých slov), *súčasný výskyt slov* (nezáleží na poradí) alebo *vzdialenosť jednotlivých slov frázy v texte*. Keďže súčasný spoločný výskyt dvoch a viacerých slov nemusí znamenať to, že tvoria frázu, nie je táto metóda stopercentne úspešná.

Ďalšou je syntaktická identifikácia – obdobná metóda ako štatistická identifikácia. Rozdielom je, že pri identifikácii fráz medzi jednotlivými slovami potenciálnej frázy je analyzovaná ich syntaktická zložka. Frázy sa najskôr vyhľadajú štatistickou identifikáciou a ich relevancia sa potvrdí existenciou záznamu v existujúcich slovníkoch fráz, prípadne riadených slovníkoch.

Eliminácia neplnovýznamových slov. Neplnovýznamové (stop) slová sú funkčné časti textu, ktoré nenesú žiaden význam. Príkladom sú napr. spojky, častice, citoslovcia... Pod nešpecifickými slovami rozumieme slová či frázy, ktoré sa vyskytujú vo väčšine dokumentov korpusu a teda majú minimálnu selektívnu silu. Oba typy slov je potrebné pomocou negatívneho slovníka (slovník stop-slov) z ďalšieho spracovania vylúčiť, pretože pri probléme vyhľadávania by do výsledku vnášali len šum.

Lematizácia. V texte sa jednotlivé slová vyskytujú v rôznych tvaroch – číslach, pádoch, časoch a pod. Preto je pre účely automatického spracovania textov potrebné realizovať ich redukciu na základný tvar, kmeň, resp. koreň a to: pomocou slovníka kmeňov a koreňov (výhodou je prakticky nulová chybovosť, nevýhodou je možná rozsiahlosť slovníka), odstránením afixov, (tzn. sufixov- prípon a prefixov - predpôn) alebo štatistickou metódou (pomocou frekvencie jednotlivých zhlukov písmen sa učí, či sa jedná o predponu, príponu alebo koreň slova). Nezanedbateľnou výhodou tejto metódy je nezávislosť od jazyka, naopak za nevýhodu je možné považovať neschopnosť rozlíšiť inflexné a derivačné afixy.

Váhovanie. Rozličné slová textového dokumentu nie sú rovnako dôležité pre reprezentáciu obsahu dokumentu. Preto je dôležité definovať relatívnu hodnotu – váhu slova, ktorá vyjadruje nakoľko dané slovo reprezentuje

obsah dokumentu. Váha reprezentuje selektívnu silu termu, ktorá jadruje schopnosť termu vyhľadať z korpusu množinu dokumentov, ktorá sa výrazne líši od množín dokumentov vyhľadaných pomocou iných termov. Vyššiu selektívnu silu majú tie termy, ktoré majú vysokú frekvenciu výskytu v konkrétnom dokumente, ale v ostatných dokumentoch kolekcie sa vyskytujú zriedkavo. Term, ktorý nájde všetky dokumenty korpusu má minimálnu selektívnu silu.

Výsledný zoznam indexovaných termov, teda slov v dokumente, je potom možné usporiadať podľa ich váhy. Táto váha sa môže použiť na redukciu počtu termov, reprezentujúcich dokument. Ponechajú sa iba tie termy, ktorých váha presahuje určitú zadanú hraničnú hodnotu. Alebo sa jednoducho vyberie určitý počet termov s najvyššou váhou.

Redukcia príznakového priestoru. V doméne klasifikácie textov sa často pracuje s dokumentmi korpusov, ktorých dimenzia príznakového priestoru nadobúda vysoké hodnoty. V súvislosti s tým narastá čas výpočtu algoritmov, použitých na spracovanie takých korpusov textových dokumentov. Preto bola vyvinutá skupina automatických metód, ktoré na základe štatistických informácií o zvolenom korpuse umožňujú redukciu jeho príznakového priestoru. Tieto metódy pracujú na princípe ohodnocovania dôležitosti termov a následného odstránenia príznakov – termov, ktorých dôležitosť je menšia ako zvolený prah. V rámci našich experimentov bola použitá metóda ohodnocovania dôležitosti termov podľa informačného zisku [Yiming-Pedersen, 1997]. **Informačný zisk** určuje množstvo obsiahnutej informácie v terme zisťovaním prítomnosti resp. neprítomnosti termu v dokumente. Majme definovanú množinu kategórií. Informačný zisk termu *t* je definovaný vzťahom

$$G(t) = -C + P + A \tag{41}$$

$$C = \sum_{i=1}^{m} \Pr(c_i) \log \Pr(c_i) \tag{42}$$

$$P = \Pr(t) \sum_{i=1}^{m} \Pr(c_i \mid t) \log \Pr(c_i \mid t) \tag{43}$$

$$A = \Pr(\bar{t}) \sum_{i=1}^{m} \Pr(c_i \mid \bar{t}) \log \Pr(c_i \mid \bar{t})$$

(44)

kde $\Pr(c) = \frac{N_c}{N}$ je pravdepodobnosť kategórie *c* a $\Pr(c \mid t) = \frac{N_{tc}}{N_t}$ je podmienená pravdepodobnosť výskytu termu *t* v kategórii *c* a napokon $\Pr(c \mid \neg t) = \frac{N_{\neg tc}}{N_{\neg t}}$ je podmienená pravdepodobnosť absencie termu *t* v kategórii *c*. Ďalej *m* reprezentuje počet kategórií, *N* je počet dokumentov, N_c je počet dokumentov kategórie c, N_t je počet dokumentov obsahujúcich term *t*, N_{tc} je počet dokumentov kategórie *c*, obsahujúcich term *t*, $N_{\neg t}$ je počet dokumentov neobsahujúcich term *t*, $N_{\neg tc}$ je počet dokumentov kategórie *c*, neobsahujúcich term *t*. Veličiny *C, P* a *A* nemajú zvláštny význam a slúžia na zjednodušenie zápisu veľmi dlhého vzťahu.

4.3 Vplyv váhovania slov na presnosť kategorizácie textov

Proces definície a výpočtu váh sa nazýva „váhovanie“. Rozličné typy váhovacích schém je možné nájsť v [Salton-Buckley, 1988]. V našich experimentoch bola hľadaná odpoveď na otázku: Akú váhovaciu schému by bolo najvhodnejšie použiť v procese predspracovania textových dokumentov, aby bola dosiahnutá efektívnejšia kategorizácia textových dokumentov v procese ich spracovania?

V rámci našich experimentov bola použitá vektorová reprezentácia dokumentov. Vektorový model pre množinu *m* dokumentov reprezentuje matica *A* veľkosti *m* x *n*, kde *n* udáva počet uvažovaných termov v rozsahu celej kolekcie dokumentov (napríklad počet termov po predspracovaní dokumentov). Prvkami matice sú váhy. (Matica *A* už bola definovaná aj v podkapitole 4.1. Na tomto mieste sa časť tejto definície opakuje kvôli nadväznosti na definície váhovacích funkcií.) Predpokladajme, že $F = A^T$. Potom váha w_{ij} termu t_j v dokumente d_i môže byť vyčíslená podľa vzťahu $w_{ij} = F(d_i, t_j)$. Funkcia *F* je takzvaná “váhovacia funkcia”. Táto funkcia môže byť definovaná rozličným spôsobom a tak môže definovať rôznu váhova-

ciu schému. V našich experimentoch boli testované nasledovné váhovacie schémy:

Binárne váhovanie. Váhovacia funkcia je $F: TxC \rightarrow \{0, 1\}$, kde C je korpus dokumentov a T je množina termov, pre ktoré F nadobúda hodnotu $F(d_i, t_j) = 1$ v prípade, že sa term t_j nachádza v dokumente d_i aspoň raz. Inak je $F(d_i, t_j) = 0$.

TF váhovanie (Term Frequency). Váhovacia funkcia je definovaná nasledovne $F: TxC \rightarrow N$, kde množina N predstavuje prirodzené čísla a $F(d_i, t_j) = k$ reprezentuje termovú frekvenciu t_j v dokumente d_i. Táto váhovacia funkcia obsahuje informácie o sile termu, získané iba z partikulárneho dokumentu. Dôležitosť termu vo vzťahu k celému korpusu dokumentov nie je uvažovaná.

TF-IDF váhovanie. Je to kombinácia predchádzajúceho TF váhovania lokálneho charakteru a globálneho váhovania IDF (Inverse Document Frequency), ktorého váhovacia funkcia je definovaná vzťahom $G(t_j) = idf_j = log(N/df_j)$, kde N je počet dokumentov v korpuse a df_i je počet dokumentov, v ktorých sa vyskytuje term t_i .

IW váhovanie (Inquery Weighting)**.** Toto váhovanie je komplikovanejšie, ale jeho výhodou je, že neobsahuje parametre, ktoré by bolo potrebné experimentálne stanoviť, ako pri nasledujúcom váhovaní Sparks, John a Robertson. Váhovacia funkcia je definovaná

$$w_{ij} = \frac{tf_{ij} \log \frac{N + 0.5}{n}}{(tf_{ij} + 0.5 + 1.5 ndl_i) \log(N + 1)},$$

(45)

kde n je počet dokumentov, ktoré obsahujú term t_i, N je celkový počet dokumentov v korpuse a ndl_j je normalizovaná dĺžka dokumentu definovaná ako pomer dĺžky dokumentu ku priemernej dĺžke všetkých dokumentov korpusu.

Váhovanie Sparck, Jones a Robertson. Toto váhovanie je vaným váhovaním, ktoré je veľmi efektívne. Jeho nevýhodou je, že niektoré parametre je potrebné experimentálne stanoviť. Hodnota váhy sa vyčísľuje podľa nasledovného vzťahu

$$w_{ij} = \frac{tf_{ij} idf_j (K1+1)}{K1(1-b+ndl_i b) + tf_{ij}} . \tag{46}$$

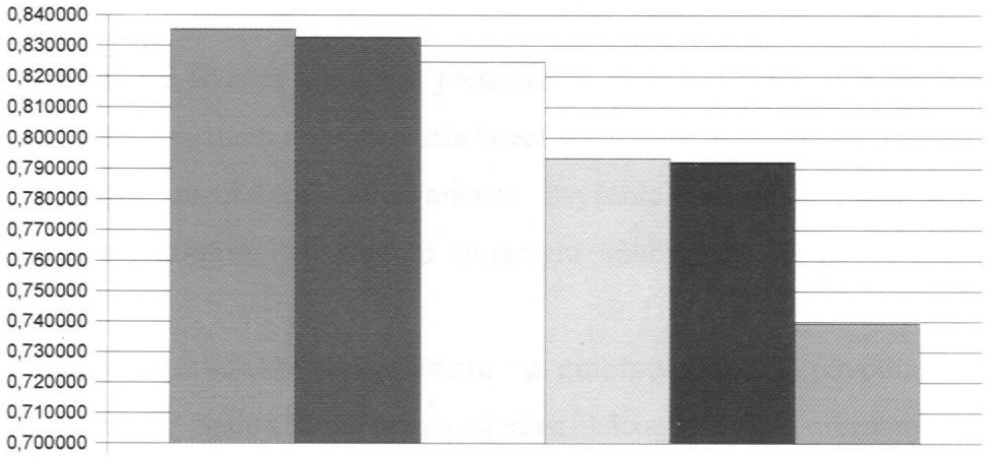
Parameter $b \in \{0,1\}$ má hodnotu *0* v prípade viactriednej klasifikácie alebo hodnotu *1* v prípade klasifikácie do jednej triedy. Parameter *K1* kontroluje vplyv frekvencie termu. Ostatné premenné sú definované tak isto, ako v predchádzajúcich váhovacích schémach.

Tab.5: Presnosť klasifikácie nad kolekciou 20 News Groups pre rôzne váhovacie schémy

	Spark&	**Inquery**	**TFIDF (ltc)**	**Binary**	**TFIDF (ntc)**	**TF**
1	0.834236	0.830225	0.822002	0.794826	0.790614	0.735058
2	0.827818	0.827016	0.818492	0.790112	0.791617	0.738969
3	0.837345	0.836141	0.828620	0.795929	0.794725	0.739571
4	0.835540	0.832130	0.824208	0.788608	0.791115	0.738468
5	0.841757	0.838448	0.830325	0.797333	0.792920	0.745187
Priemerná presnosť	0.835339	0.832792	0.824729	0.793361	0.792198	0.739450
Štandard. odchýlka	0.005075	0.004572	0.004824	0.003797	0.001653	0.003654
Max. presnosť	0.841757	0.838448	0.830325	0.797333	0.794725	0.745187
Min. presnosť	0.827818	0.827016	0.818492	0.788608	0.790614	0.735058
Usporiadanie	**1**	**2**	**3**	**4**	**5**	**6**
%	100.0	99.7	98.7	95.0	94.8	88.5

Všetky uvedené váhovacie schémy boli testované na dvoch kolekciách dokumentov: Reuter's 21578 a 20 News Groups, ktorá obsahovala 19953 dokumentov priradených do práve jednej z dvadsiatich kategórií. Dimenzia lexikálneho profilu bola 111474. Pre účely testovania bola táto množina náhodným výberom rozdelená v pomere 1:1 na trénovaciu a testovaciu časť. Výsledky získané nad kolekciou dokumentov 20 News Groups sú

uvedené v tab.5 a preferujú váhovaciu schému Sparck, Jones & Robertson (SJR). Výsledky experimentov z tab.5 sú znázornené prehľadnejším spôsobom na obr.13. Podrobnejšie informácie o testoch s váhovaním je možné nájsť v [Machová-Maták-Bednár, 2005].



Obr.13: Priemerná presnosť klasifikácie na kolekcii 20 News Groups v závislosti od použitej váhovacej schémy (zľava do prava: Sparck, Jones & Robertson, Inquery, TFIDF(ltc), binary, TFIDF(ntc) a TF)

4.4 Generovanie kľúčových slov

V oblasti spracovania textových dokumentov sa často vyskytujú dokumenty vyznačujúce sa veľkou dimenziou lexikálneho profilu. To znamená, že vo vektorovej reprezentácii dokumentov sa aj po predspracovaní nachádza príliš veľa slov – termov. To je dôvod, prečo boli vyvinuté štatistické metódy na redukciu lexikálneho profilu (napríklad informačný zisk, vzájomná informácia, χ^2 štatistika). Tieto metódy sa môžu použiť aj na generovanie kľúčových slov textových dokumentov. Na účely generovania kľúčových slov je možné použiť aj váhovaciu funkciu, napríklad TF-IDF. Všetky tieto metódy hodnotia dôležitosť termu. Termy s menšou váhou ako je zvolená hraničná hodnota sa vylúčia z lexikálneho profilu. Tie termy, ktoré ostanú môžu byť kľúčové slová daného textového dokumentu pri vhodne nastavenej hraničnej hodnote váhy slova, resp. termu.

Informačný zisk (Information Gain). Je to štatistická metóda [Yiming-Pedersen, 1997], ktorá sa často používa v oblasti strojového učenia na určenie dôležitosti termu. Informačný zisk reprezentuje kvantitatívnu informáciu o dôležitosti termu. Jeho výpočet bol v tejto práci uvedený v podkapitole „Predspracovanie textových dokumentov“, konkrétnejšie v podkapitole „Redukcia príznakového priestoru“.

Vzájomná informácia (Mutual Information). Metóda vzájomnej informácie [Yiming-Pedersen, 1997] je štatistická metóda zvyčajne používaná v štatistických jazykových modeloch, ktoré modelujú vzťahy medzi slovami (termami). Predpokladajme kontingenčnú tabuľku termu *t* a kategórie *c*, kde *A* je súčasný počet výskytov termu *t* a kategórie *c*; B je počet výskytov termu *t* mimo kategórie *c*; *C* je počet výskytov kategórie *c* mimo termu *t* a *N* je počet všetkých dokumentov. *I(t,c)* má hodnotu nula, keď term *t* a kategória *c* sú od seba nezávislé. Potom je možné vzájomnú informáciu termu *t* a kategórie *c* vyčísliť nasledovne

$$I(t,c) = \log \frac{\Pr(t \wedge c)}{\Pr(t) \times \Pr(c)} \tag{47}$$

a teda

$$I(t,c) \approx \log \frac{A \times N}{(A+C) \times (A+B)} . \tag{48}$$

χ^2 štatistika [Yiming-Pedersen, 1997] je štatistická metóda, ktorá určuje mieru závislosti termu *t* a kategórie c. Predpokladajme kontingenčnú tabuľku s hodnotami *A, B, C* a *N* definovanými v predchádzajúcom odseku. Potom χ^2 štatistiku termu *t* môžeme vyčísliť pomocou nasledovného vzťahu

$$\chi^2(t,c) = \frac{N \times (AD - CB)^2}{(A+C) \times (B+D) \times (A+B) \times (C+D)} . \tag{49}$$

Tento vzťah nadobúda nulovú hodnotu, keď term *t* a kategória *c* sú vzájom nezávislé, podobne ako v pri metóde vzájomnej informácie. χ^2 štatistika sa počíta pre každý pár term – kategória v rámci celého korpusu dokumentov. Potom sa počíta zložené skóre dvoma alternatívnymi spôsobmi

$$\chi^2_{avg}(t) = \sum_{i=1}^{m} \Pr(c_i)\chi^2(t, c_i) \tag{50}$$

$$\chi^2_{\max}(t) = \max_{i=1}^{m}\{\chi^2(t, c_i)\} \tag{51}$$

Hlavný rozdiel medzi χ^2 štatistikou a metódou vzájomnej informácie je fakt, že χ^2 štatistika je normovaná hodnota. Preto táto metóda nedáva dobré výsledky pre termy s nízkou frekvenciou výskytu.

TF-IDF váhovanie [Salton-Buckley, 1988] vyberá charakteristické slová z množiny dokumentov na báze výpočtu tf-idf váh pre všetky slová – termy z danej množiny dokumentov. Termy s váhou vyššou ako zvolená hraničná hodnota tvoria množinu charakteristických slov V_g. Hraničnú hodnotu spravidla určuje používateľ podľa požadovaného počtu termov v množine V_g. Táto množina je orezávaná pomocou množiny zakázaných slov V_z definovanej používateľom alebo systémovým manažérom. Je to množina slov majúca všeobecný význam alebo sú to slová neobsahujúce podstatné informácie. Potom konečná množina kľúčových slov sa určí nasledovne $V=V_g - V_z$.

4.4.1 Experimenty s generovaním kľúčových slov

Bola vykonaná séria testov nad kolekciou dokumentov 20 News Groups. Z tejto kolekcie boli generované charakteristické slová pre každú z dvadsiatich kategórií. Tieto slová boli manuálne posudzované, či môžu plniť úlohu kľúčových slov. Takýmto spôsobom boli generované kľúčové slová pomocou každej z uvedených štyroch metód: informačný zisk, vzájomná informácia, χ^2 štatistika a TF-IDF metóda. Kandidáti na pozíciu

kľúčových slov v piatich zvolených kategóriách z celkového počtu dvaddvadsať, ktorí boli generovaní metódou informačného zisku sú uvedení v tab.6.

Postup použitý na generovanie kľúčových slov bol nasledovný. Po predspracovaní bol vyčíslený informačný zisk každého termu kolekcie dokumentov. Podľa hodnoty informačného zisku boli termy usporiadané. Z usporiadaného zoznamu termov bolo vybratých 16 najlepších s najvyšším informačným ziskom. Títo šestnásti kandidáti kľúčových slov boli „manuálnym výberom“ rozdelení do nasledovných skupín:

- skupina termov, ktoré je možné považovať za kľúčové slová (v tab.6, tab.7, tab.8 a tab.9 sú označené tučným písmom),
- skupina slov, ktoré sú zaujímavé, ale nie sú kľúčové (v tab.6 až tab.9 sú označené kurzívou),
- skupina slov, ktoré nie sú kľúčovými slovami, ani nie sú zaujímavé.

Metóda Informačného zisku (tab.6) sa zdá byť vhodnou metódou na generovanie kľúčových slov s akceptovateľnou mierou presnosti, ktorá je daná počtom získaných kľúčových slov v pomere ku všetkým generovaným slovám.

Kandidáti kľúčových slov, získaní metódou vzájomnej informácie sú uvedení v tab.7. Výsledky získané touto metódou sú oveľa horšie ako v predchádzajúcom prípade a konečný počet kľúčových slov je minimálny.

Tab.6: Kandidáti kľúčových slov generovaných z kolekcie 20 News Groups metódou informačného zisku

	01.atheism	02.comp.graphics	08.rec.auto	12.sci.crypt	15.sci.space
1	writes	**Graphics**	**car**	**Key**	**Space**
2	article	**Image**	**cars**	**Encryption**	**Orbit**
3	**god**	*Program*	**Engine**	Clipper	**Nasa**
4	don't	*File*	**Ford**	*Chip*	**Earth**
5	people	*Files*	Article	**government**	Shuttle
6	**religion**	**Images**	Dealer	**Keys**	Launch
7	keith	**Format**	*Miles*	Writes	Writes
8	point	*Computer*	**Auto**	*Public*	Pat
9	fact	Code	**Drive**	**Security**	**Moon**
10	**atheists**	ftp	**Price**	**System**	Henry
11	wrote	**Color**	**Driving**	Nasa	Spencer
12	objective	**Software**	Good	**Secure**	**Solar**
13	claim	**Gif**	Buy	**Escrow**	Project
14	*moral*	*Version*	**Speed**	*Algorithm*	*Mission*
15	jon	Email	**Toyota**	**information**	Cost
16	*morality*	Advance	**Oil**	Pgp	*Flight*

Tab.7: Kandidáti kľúčových slov generovaných z kolekcie 20 News Groups metódou vzájomnej informácie

	01.atheism	02.comp.graphics	08.rec.auto	12.sci.crypt	15.sci.space
1	writes	Writes	Writes	Writes	Writes
2	article	Article	Article	Article	Article
3	don't	Don't	don't	don't	don't
4	people	i'm	i'm	People	Time
5	i'm	Time	Good	i'm	it's
6	time	it's	it's	it's	People
7	it's	Good	People	Time	i'm
8	good	People	Time	Make	Make
9	make	Find	Make	Good	Good
10	point	Make	***Car***	System	Work
11	doesn't	i've	i've	Work	**Space**
12	things	Work	Back	**government**	Find
13	**god**	University	Work	**information**	Things
14	fact	Problem	problem	Find	System
15	can't	Information	can't	can't	Years
16	thing	*Program*	Thing	Point	Back

Treťou metódou je χ^2 štatistika. Použitá procedúra výberu kandidátov kľúčových slov bola podobná ako pri predchádzajúcich štatistických metódach. Prvých 16 kandidátov kľúčových slov sa nachádza v tab.8. Dosiahnuté výsledky sú porovnateľné s tými získanými metódou Informačného zisku. Zvlášť bola táto metóda úspešná v kategórii “computer graphics”.

Tab.8: Kandidáti kľúčových slov generovaných z kolekcie 20 News Groups metódou χ^2 štatistika

	01.atheism	02.comp.graphics	08.rec.auto	12.sci.crypt	15.sci.space
1	**atheists**	**Graphics**	**car**	**Encryption**	**Space**
2	**atheism**	**Image**	**cars**	Clipper	**Orbit**
3	livesey	**Images**	**engine**	**Key**	Shuttle
4	*benedikt*	**Gif**	**ford**	**Keys**	Launch
5	keith	**Animation**	**toyota**	**Escrow**	**Nasa**
6	o'dwyer	**Jpeg**	**mustang**	Nsa	**Spacecraft**
7	**atheist**	**Polygon**	**auto**	**Crypto**	**Moon**
8	beauchaine	**Format**	dealer	*Chip*	**Solar**
9	mathew	**Tiff**	callison	**Encrypted**	Henry
10	*morality*	Pov	*taurus*	Sternlight	Spencer
11	jaeger	**Polygons**	**nissan**	**cryptogrphy**	**Lunar**
12	**god**	**Viewer**	eliot	**Secure**	**Orbital**
13	mozumder	**Formats**	**chevy**	Pgp	**Satelite**
14	gregg	**Texture**	**engines**	*Privacy*	*Flight*
15	objective	**Tga**	**tires**	*Algorithm*	*Mission*
16	schneider	*Files*	**wagon**	**Wiretap**	**Sky**

Posledná bola použitá metóda TF-IDF. Na rozdiel od predchádzajúcich metód pri tejto metóde nebol nevyberaný presne určený počet slov usporiadaných podľa hodnôt štatistickej funkcie. V metóde TF-IDF sa najprv vypočítala váha pre každý term – slovo. Potom používateľ určil minimálnu a maximálnu hraničnú hodnotu váhy slova. Vyberali sa kandidáti kľúčových slov s váhou TF-IDF z intervalu od minimálnej po maximálnu zvolenú hraničnú hodnotu. Maximálnu hraničnú hodnotu bolo potrebné definovať preto, lebo slová s veľmi vysokou váhou boli spravidla príliš všeobecné. Interval hraničných váh je rôzny pre každú kategóriu.

Tab.9: Kandidáti kľúčových slov generovaných z kolekcie 20 News Groups metódou TF-IDF

Kategórie	Kľúčové slová
01.alt.atheism $w_t \in (3; 4)$	*black*, **god, islam, jesus, souls, dogma, lucifer, satanists, rushdie, mary, israel, messiah, isaiah, religously, crucified**
02.comp.graphicss $w_t \in (4; 5)$	**volume, quality, row,** *file*, **ray, images, gif, processing, transformations, mirror, colorview**
08.rec.autos $w_t \in (2{,}5; 3{,}5)$	**bolsters, car, inflammatory, oil, indicators, fuels,** *probe*, **diesel, gasoline,** *socket*, **diameter, abs, radar, brake, chevrolet, alarm,** *sensor*, **emissions, rotor, clunker, clutch, autobahn, carburetor, gtz,** *sprint*, **braking, ethanol, skidpad, carerra, idling, diesels, diaphram, overboost, vehical**
12.sci.crypt $w_t \in (2{,}5; 2{,}9)$	*detection, networking*, **ansi,** *wordperfect*, **symbolic, encryption, passwords, cryptanalysis, cryptanalyst, cypherpunks, keyphrase, cryptosystem, coder**
15.sci.space $w_t \in (2{,}5; 3)$	**universe, moon, atmosphere, landscape,** *physicist*, **planets, solar, nasa, ship, comet, astronomical, explorer, sun, infrared, spacecraft, orbiter,** *detectors*, **ozone, saturn, mercury, asteroids, astronaut, martian, rocketry, neptune, constellation**

Tab.9 ilustruje kandidátov kľúčových slov pre 5 zvolených kategórií kolekcie 20 News Groups generovaných metódou TF-IDF. Použitím tejto metódy bol získaných menší počet kandidátov s väčšou hodnotou váhy.

Výsledky experimentov v [Machová-Szabóová-Bednár, 2007] a v [Machová-Bednár-Mach, 2007] ukázali, že metóda vzájomnej informácie sa na riešenie problému generovania kľúčových slov dokumentu nehodí. Metódy Informačného zisku a χ^2 štatistika dávali približne rovnaké výsledky ale najsľubnejšia bola metóda TF-IDF.

Keďže metódou χ^2 štatistika a TF-IDF bolo získaných najviac kľúčových slov, výsledky týchto metód boli podrobené ďalšiemu spracovaniu, teda hľadaniu podstatných vzťahov medzi termami na základe podmienenej pravdepodobnosti výskytu termov. Na základe týchto vzťahov boli generované dvojslovné frázy.

4.4.2 Detekcia vzťahov medzi termami

Podstatné vzťahy medzi termami boli detekované na základe podmienenej pravdepodobnosti výskytu termov. Ak sa povedzme dva termy vyskytujú štatisticky významne spolu v množine dokumentov, dá sa predpokladať, že majú aj významovo k sebe blízko.

Pre každý pár termov $\{(t_i,t_j), t_i, t_j \in V\}$ bol určený počet dokumentov o_{ij} v ktorých sa oba termy nachádzali súčasne [Jelínek, 2005]. Následne bola určená podmienená pravdepodobnosť pre každý term t_i

$$p_{i|j} = \frac{o_{ij}}{n_j}, i \neq j \tag{52}$$

kde n_j je počet dokumentov, v ktorých sa vyskytuje term t_j. Táto hodnota bola použitá na rozlišovanie nasledovných typov vzťahov, v ktorých m je vopred stanovená medza minimálneho uvažovaného výskytu termov:

($p_{i|j}$ > m) $\wedge$ ($p_{j|i}$ < m) – term t_i sa vyskytuje vo väčšom počte dokumentov ako term t_j. Term t_i je teda všeobecnejší ako term t_j.

($p_{i|j}$ < m) $\wedge$ ($p_{j|i}$ > m) – term t_i sa vyskytuje v menšom počte dokumentov ako term t_j. Term t_i je teda špecifickejší ako term t_j.

($p_{i|j}$ > m) $\wedge$ ($p_{j|i}$ > m) – termy t_i a t_j sa vyskytujú často spolu a ich vzájomný vzťah je silný a vyvážený.

($p_{i|j}$ < m) $\wedge$ ($p_{j|i}$ < m) – relácia medzi termami t_i and t_j je slabá. Ich súčasný výskyt v dokumentoch je skôr náhodný.

Dá sa predpokladať, že pojem, ktorý sa vyskytuje vo väčšom počte dokumentov z rôznych kategórií, bude všeobecnejší (napríklad „potom, alebo“ kontra „klasifikátor“). Zriedkavý spoločný výskyt dvoch termov vo väčšine prípadov znamená, že ich súčasný výskyt je skôr náhodný (spolu sa vyskytujú oveľa zriedkavejšie ako samostatne). Zriedkavé ale ustálené slovné spojenie je iný prípad (vyskytujú sa takmer vždy spolu).

4.4.3 *Experimenty s generovaním dvojslovných fráz*

Testy boli uskutočnené nad kolekciou textových dokumentov 20 News Groups. Dvojslovné frázy boli formované z kľúčových slov získaných metódou χ^2 statistics a TF-IDF. Získané páry termov boli rozdelené na:

- frázy (tučným písmom zvýraznené slová v tab.10 a tab.11),
- páry termov nachádzajúce sa často spolu, majúce významovú závislosť (viď tab.10 a tab.11),
- páry termov nachádzajúce sa často bez významovej závislosti.

Tab.10: Páry termov – dvojslovné frázy generované z kolekcie 20 News Groups metódou χ^2 štatistika

Kategórie	Páry termov
01.alt.atheism	atheists-atheism, morality-objective, morality-moral, objective-moral
02.comp.graphicss	gif-tiff, gif-formats, jpeg-tiff, polygons-texture, polygons-vertices, program-file, ***adobe-photoshop***
08.rec.autos	mustang-taurus, mustang-camaro, callison-camaro, ***chevy-camaro***, ***sedan-wagon***
12.sci.crypt	***encryption-key***, encryption-chip, encryption-cryptography, encryption-secure, encryption-privacy, ***encryption-algorithm***, encryption-communications, ***encryption-scheme***, cryptography-privacy, wire-tap-phones, decrypt-encrypt
15.sci.space	orbit-shuttle, orbit-launch, orbit-moon, orbit-solar, orbit-satellite, orbit-mission, shuttle-nasa, shuttle-flight, shuttle-mission, payload-missions, spacecraft-satellites, ***spacecraft-propulsion***, spacecraft-mars, spacecraft-missions, moon-lunar, ***henry-spencer***, lunar-mars, orbital-propulsion, satellites-missions, mars-spacecraft, mars-missions, mars-jupiter, jupiter-orbiting

Pre ilustráciu sú v tab.10 uvedené dvojslovné frázy generované metódou χ^2 štatistika a v tab.11 sa nachádzajú frázy odvodené pomocou TF-IDF váhovacej schémy. Porovnaním výsledkov prezentovaných v týchto dvoch tabuľkách vyplýva, že lepšie výsledky boli získané metódou TF-IDF, lebo táto metóda generuje viac dvojslovných fráz [Machová-Szabóová, 2007].

Tab.11: Páry termov – dvojslovné frázy generované z kolekcie 20 News Groups metódou TF-IDF

Kategórie	Páry termov
01.alt.atheism	god-jesus, moral-objective, islam-rushdie, mary-israel, mary-messiah, israel-messiah, israel-crucified, isaiah-messiah
02.comp.graphicss	***volume-processing***, volume-transformations, ***quality-processing***, quality-sgi, row-colorview, ray-mirror, images-gif, quantitative-transformations, gif-images, processing-sgi, sgi-quality, ***mirror-transformations***, mirror-colorview
08.rec.autos	***alternative-fuels***, alternative-substitutes, bolsters-clumsy, fluid-temperature, indicating-diameter, indicating-lockup, indicating-ethanol, diesel-gasoline, diesel-emissions, ***abs-braking***, brake-clutch, alarm-sensor
12.sci.crypt	***technology-encryption***, accounts-passwords, ***message-encryption***, functions-cryptanalysis, functions-cryptosystem, regular-passwords, regular-cypherpunks, chip-encryption, symbolic-fields, passwords-usage, cryptanalysis-cryptosystem
15.sci.space	***comprehensive-fusion***, universe-theory, data-solar, data-nasa, data-spacecraft, moon-solar, atmosphere-planets, tools-sophisticated, landscape-volcanic, landscape-neptune's, landscape-craters, planets-orbiter, planets-saturn, planets-mercury, detectors-cloud, ozone-observer, saturn-mercury, asteroids-fusion, martian-observer

4.4.4 Experimenty s kolekciou Times

Boli uskutočnené aj podobné experimenty nad kolekciou Times, ktorej dokumenty neboli zatriedené vopred do kategórií. Preto bolo potrebné dokumenty pred generovaním kľúčových slov zhlukovať pomocou zhlukovacej techniky k-centier. Avšak výsledky získané z kolekcie 20 News Groups boli lepšie - presnejšie ako výsledky generované z kolekcie Times. V kolekcii 20 News Groups sa našlo viac kľúčových slov. Príčinou menej presných výsledkov kolekcie Times môže byť aj to, že dokumenty tejto kolekcie neboli indexované manuálne, ale automaticky podľa výsledku zhlukovania. V dôsledku toho mohli byť mnohé dokumenty zaradené do nesprávnej kategórie.

Pre Ilustráciu sú v tab.12 uvedení kandidáti kľúčových slov, ktorí boli získaní z kolekcie Times metódou informačného zisku, ktorá dávala pre túto kolekciu najlepšie výsledky. Pred začiatkom generovania kľúčových slov boli dokumenty kolekcie zhlukované postupne do jednej až siedmych tried. Experimentálne bolo určené, že optimálny počet zhlukov je sedem.

Tab.12: Kandidáti kľúčových slov generovaní z kolekcie Times metódou informačného zisku

Zhluky	Kľúčové slová
1	week, **market, nato**, common, *de, west,* **europe,** *nuclear,* **british, germany,** *minister,* **britain, britain's,** *charles,* **economic***, state,* **france, french***, western, gaulle*
2	**russia,** party, day, *nikita*, **job,** cent, **committee***, khrushchev,* year's, **chairman, soviet,** authorities, **officials,** made**, production, factory,** miles, **boss, river,** lost
3	**communist,** car**, west,** showed, **paris, wall, office,** ancient, **vote, dropped,** dark, **charged, history, germany,** found, weather, hour, good, **charges, parliamentary**
4	**moscow, russian, work,** years, **intellectuals,** ago, **leningrad,** author, **police,** began, **history,** writers*,* ***soviet,*** hands, months, **agents,** talk, *ranging,* turned, open
5	began, **troops, congo, government,** back, **communist,** hours, set, **soldiers, katanga,** miles, **central, moise, african,** men, *town*, **killed,** *streets*, **armed, embassy**
6	**leaders, government,** week's**, political, army, king, arab, defense,** long, years, city, people, party, **premier, nation, moslem,** *cabinet,* ii, **nasser,** year
7	**india's, india, capital,** high, **indian, china, politicians, nehru, independence,** years, **services, prime, military, nation's, pakistan,** turn**, movement,** led**, egypt, president**

Kľúčové slová získané pre sedem zhlukov kolekcie Times boli podrobené ďalšiemu spracovaniu za účelom detekcie dvojslovných fráz založenej na podmienenej pravdepodobnosti, podobne ako pri kolekcii 20 News Groups. Páry termov generované z kolekcie Times metódou TF-IDF sú uvedené v tab.13.

Výsledky týchto experimentov nesú informáciu, že výkonnosť metódy závisí od charakteristík kolekcie dokumentov.

Tab.13: Páry termov – dvojslovné frázy generované z kolekcie 20 News Group metódou TF-IDF

Zhluky	Páry termov
1	***market - west***, market - europe, ***market - germany***, ***market - britain***, ***market - france***, ***nato - west***, ***nato - europe***, ***germany - nato***, germany - minister, germany - britain, ***germany - economic***, germany - france, ***britain - nato***, britain - minister, ***britain - economic***, ***charles - de***, ***charles - gaulle***, france - germany, france - minister, france - britain, ***france – economic***
2	***russia - nikita***, ***russia - khrushchev***, ***russia - soviet***, russia - made, ***nikita - committee***, ***nikita - khrushchev***, ***nikita - soviet***, job - boss, ***job - lost***, cent - day, cent - officials, ***cent - made***, ***cent - lost***, ***committee - khrushchev***, ***committee - chairman***, ***khrushchev - soviet***, ***soviet - officials***, made - soviet, ***factory – production***
3	***communist - west***, west - paris, ***west - germany***, paris - germany, ***vote - parliamentary***
4	***moscow - russian***, moscow - work, ***moscow - soviet***, ***russian - soviet***, ***russian - agents***, ***intellectuals - writers***, intellectuals - ranging, ***police - work***, soviet – work
5	***troops - government***, troops - communist, ***troops - soldiers***, ***troops - central***, ***troops - killed***, ***troops - armed***, ***congo - soldiers***, ***congo - katanga***, ***congo - moise***, ***government - communist***, ***soldiers - killed***, ***soldiers - armed***, ***soldiers - embassy***, ***katanga - moise***, town - soldiers, ***town – embassy***
6	***leaders - government***, ***leaders - army***, leaders - premier, ***leaders - nation***, ***leaders - cabinet***, ***leaders - nasser***, ***government - political***, ***government - army***, ***government - defense***, government - premier, ***government - nation***, government - cabinet, ***political - leaders***, political - army, political - defense, ***army - defense***, army - nation, ***king - arab***, ***king – nasser***, ***arab – Nasser***
7	***india - indian***, ***india - china***, ***india - nehru***, ***india - pakistan***, ***india - egypt***, capital - independence, ***capital - military***, ***high - independence***, ***china - military***, ***politicians - nation's***, ***nehru - india***, ***nehru - pakistan***, military - president, ***movement – Egypt***

4.5 Zhlukovanie textových dokumentov

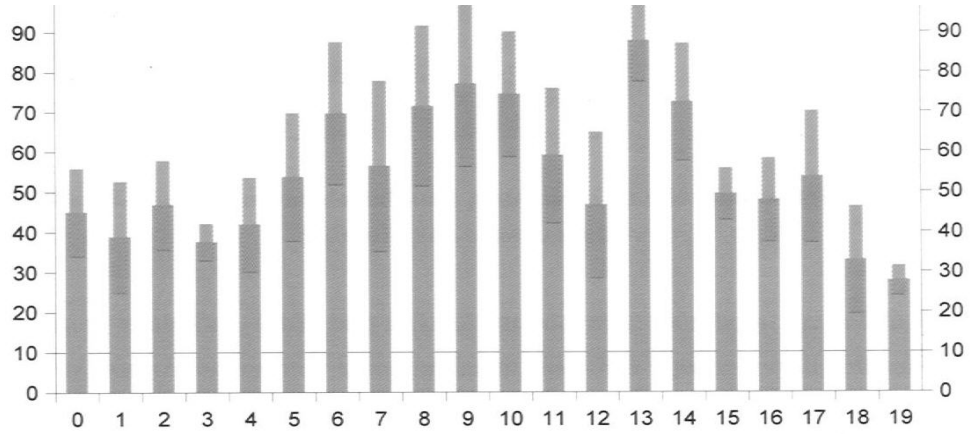
Zhlukovanie je proces zoskupovania objektov (trénovacích príkladov), opísaných ich vlastnosťami (hodnotami atribútov, resp. príznakov), do zhlukov na základe podobnosti. Príznak môže byť atribút, alebo kombinácia atribútov. Jedna z možností reprezentácie znalostí je trojica objekt-atribút-hodnota. Na druhej strane, v klasifikácii sa používajú príznakové metódy, kde klasifikátor rozhoduje na základe hodnoty príznakov. Pri zhlukovaní dokumentov boli ako príznaky použité všetky slová dokumentov, ktoré zostali po ich predspracovaní (vylúčenie neplnovýznamových slov, lematizácia, váhovanie) a po redukcii príznakového priestoru metódou informač-

ného zisku na cca 10% pôvodnej veľkosti lexikálneho profilu (z 111474 na 11148 dokumentov).

Podobnosť objektov bola spravidla vyjadrená vzdialenosťou objektov v priestore pojmov, resp. priestore príznakov. Tento proces sa uskutočňoval bez akejkoľvek apriórnej informácie o triedach týchto objektov, preto predstavoval nekontrolovanú techniku strojového učenia. Naše experimenty sa týkali zhlukovania dokumentov [Muresan-Harper, 2001] a aplikovaný bol algoritmus k-centier, ktorý bol v tejto práci opísaný v podkapitole 2.4.2 „Zhlukovacie algoritmy strojového učenia".

4.5.1 Experimenty so zhlukovaním s náhodnou inicializáciou

Bola realizovaná množina desiatich experimentov so zhlukovacou metódou k-centier nad kolekciou textových dokumentov 20 News Groups. Každý z desiatich experimentov štartoval s inou skupinou náhodne vygenerovaných čísel, pričom každému číslu bol priradený dokument z korpusu s príslušným poradovým číslom – teda náhodné centrum zhluku. Tieto centrá tvorili inicializačnú množinu centier zhlukovania. Počet zhlukov a teda aj centier bol prirodzene stanovený na dvadsať, keďže korpus 20 News Groups obsahoval 20 kategórií dokumentov. Pri tejto náhodnej inicializácii by sa mohlo teoreticky stať, že všetky centrá by boli vybraté z tej istej kategórie. S týmto problémom sa dokáže vysporiadať „zhlukovanie s kontrolovanou inicializáciou", ktoré bude ďalej opísané. Ukončovacia podmienka bola stanovená ako logická disjunkcia dvoch rôznych kritérií. Prvého vo forme maximálneho počtu iterácií rovného hodnote 50 a druhého vo forme rozdielu medzi dvoma nasledovnými hodnotami chybovej funkcie - epsilon (stanovené na 0,1). Všetky dokumenty korpusu boli váhované pomocou Sparck, Jones & Robertson (SJR) váhovacej funkcie, ktorá bola definovaná v podkapitole 4.3 a v rámci uskutočnených experimentov sa vyznačovala najvyššou efektívnosťou. Konkrétne výsledky testov sú ilustrované na obr.14.



Obr.14: Priemerná presnosť (hrubší stĺpec) a štandardná odchýlka (tenší stĺpec) zhlukovania s náhodnou inicializáciou pomocou k-centier v závislosti od 20-tich kategórií kolekcie 20 News Groups (číslované od 0 do 19)

Taktiež sú uvedené v tab.14. V ideálnom prípade, pri stopercentnej presnosti by boli dokumenty konkrétnej kategórie zaradené do toho istého zhluku, čo znamená, že štandardná odchýlka by bola nulová. V našich experimentoch so zhlukovaním s náhodnou inicializáciou [Machová-Maták-Bednár, 2005] sa štandardná odchýlka pohybovala v intervale <3.83,21.39> %, pričom presnosť výrazne závisela od inicializácie centier.

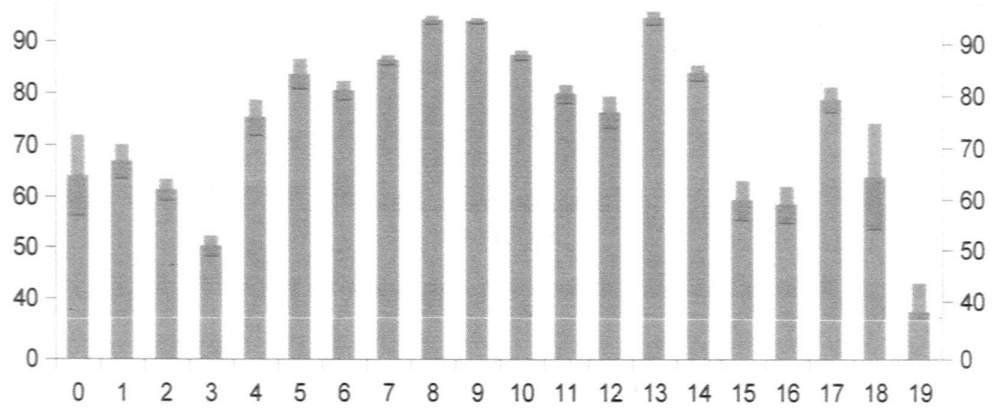
4.5.2 Experimenty so zhlukovaním s kontrolovanou inicializáciou

Bola vykonaná aj parametricky identická séria experimentov k experimentom opísaným vyššie. Jediným rozdielom bolo, že inicializácia centier nebola náhodná ale kontrolovaná. To znamená, že generovanie **i**-teho zhluku bolo inicializované centrom – príkladom - dokumentom, ktorý reprezentoval priemer desiatich náhodne zvolených príkladov z **i**-tej kategórie. Výsledky týchto experimentov [Machová-Maták-Bednár, 2006] ukázali zmysel a efektívnosť kontrolovanej inicializácie, ktorá znížila štandardnú odchýlku presnosti v porovnaní s náhodnou inicializáciou. Štandardná odchýlka sa pri zhlukovaní s kontrolovanou inicializáciou pohybovala v intervale <0.59,7.86> %.

Tab.14: Priemerná presnosť a štandardná odchýlka zhlukovania algoritmom K-centier s kontrolovanou inicializáciou, ako aj bez nej

Zhlukovanie	**bez kontrolovanej Inicializácie**		**s kontrolovanou Inicializáciou**	
Kategória	**Presnosť**	**Štandardná Odchýlka**	**Presnosť**	**Štandardná Odchýlka**
1	44,96	10,96	64,09	7,86
2	38,87	13,94	66,91	3,37
3	46,80	11,11	61,40	2,14
4	37,49	04,69	50,40	1,89
5	41,89	11,78	75,44	3,47
6	53,68	16,09	83,81	2,92
7	69,61	17,94	80,69	1,88
8	56,48	21,39	86,62	1,00
9	71,44	20,13	94,41	0,85
10	77,04	20,96	94,20	0,59
11	74,43	15,72	87,62	0,93
12	59,01	16,96	80,23	1,81
13	46,65	18,27	76,72	3,07
14	87,78	10,11	94,94	1,41
15	72,51	14,82	84,35	1,60
16	49,43	06,54	59,76	3,84
17	47,87	10,42	58,98	3,50
18	53,69	16,55	79,22	2,51
19	32,87	13,49	64,35	10,41
20	27,66	03,83	37,89	5,79

Obr.15 ilustruje tieto výsledky, ktoré sú uvedené aj v tab.14. Ako bolo uvádzané, naše experimenty boli vykonané nad kolekciou textových dokumentov 20 News Groups, ktorá pozostávala z dokumentov internetových diskusií (19953 dokumentov klasifikovaných do dvadsiatich kategórii; dimenzia lexikálneho profilu bola 111474). Jej výhodou bola takmer rovnomerná distribúcia dokumentov do kategórií a implicitné zaradenie každého dokumentu do jednej kategórie. V procese zhlukovania bola informácia o kategórii ignorovaná. Ale vo fáze testovania sa táto informácia použila za účelom vyčíslenia presnosti výsledných zhlukov. Taktiež bola použitá pri kontrolovanej inicializácii centier.



Obr.15: Priemerná presnosť (hrubší stĺpec) a štandardná odchýlka (tenší stĺpec) zhlukovania s kontrolovanou inicializáciou pomocou k-centier v závislosti od 20 tich kategórií kolekcie 20 News Groups (číslované od 0 do 19)

Inicializáciu počtu zhlukov pre neznámu kolekciu je možné automatizovať viacerými spôsobmi. Jedna možnosť je vykonať experimenty s rôznym počtom zhlukov, napríklad systematicky z intervalu od 1 do 50, vyhodnotiť tieto testy a vybrať najvhodnejší počet zhlukov. Iná možnosť je začať od stredu intervalu, alebo adaptovať počet zhlukov.

4.6 Aktívne učenie

Aktívne učenie sa zameriava na vplyv predikcie kategórie neoznačkovaných trénovacích príkladov na presnosť klasifikácie. Technika aktívneho učenia sa môže s úspechom použiť, keď je síce k dispozícii veľký objem trénovacích príkladov, ale iba veľmi malá časť z nich je označkovaná, teda obsahuje aj informáciu o triede - kategórii, do ktorej trénovací príklad patrí. Preto klasifikačné metódy je možné použiť iba nad malou časťou trénovacej množiny. Klasifikačné algoritmy vygenerované z tejto malej množiny sa následne použijú na klasifikáciu zatiaľ neoznačkovaných trénovacích príkladov. Tým sa získa nová kvalitnejšia a početnejšia trénovacia množina, z ktorej sa vygenerujú nové klasifikačné pravidlá s vyššou presnosťou klasifikácie, ako ukázali experimenty.

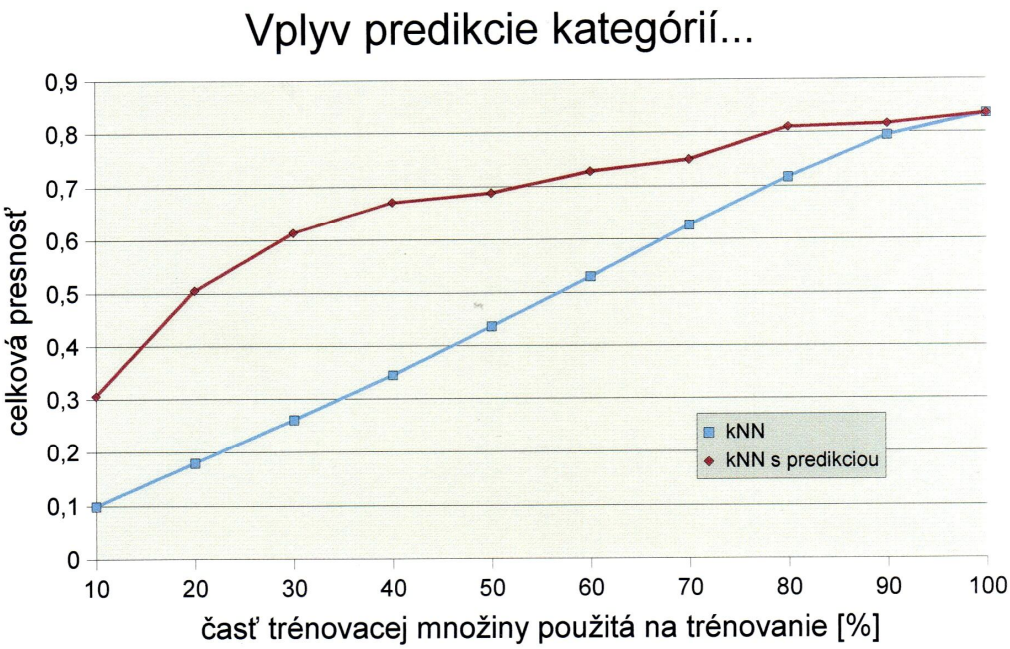
Bolo vykonaných desať experimentov v dvoch módoch. Prvý mód je založený na meraní celkovej klasifikačnej presnosti použitím úplnej trénova-

cej množiny. Druhý mód reprezentuje prípad, keď kategória zvyšných *(100% - i*10%)* dokumentov z trénovacej množiny bola predikovaná použitím kNN klasifikátora, ktorý bol natrénovaný na iba *i*10%* časti označkovaných dokumentov z trénovacej množiny, kde *i* predstavuje i-tý experiment.

Tab.15: Vplyv predikcie kategórie na presnosť klasifikácie

Trénovanie [%]	**Predikcia [%]**	**Presnosť kNN**	**Presnosť kNN s predikciou**
10	90	0.0991280	0.3059036
20	80	0.1795129	0.5051619
30	70	0.2602987	0.6137115
40	60	0.3444923	0.6706425
50	50	0.4364037	0.6879824
60	40	0.5294177	0.7281748
70	30	0.6262404	0.7499248
80	20	0.7164478	0.8104641
90	10	0.7942267	0.8159767
100	0	0.8353212	0.8353212

Inak povedané, v reálnych podmienkach sa často stáva, že sú dostupné informácie o triede trénovacích príkladov iba pre časť (i*10% dokumentov) trénovacej množiny. V týchto prípadoch je často príliš obtiažne alebo jednoducho nemožné získať informáciu o triede pre zvyšnú neoznačkovanú časť (100% - i*10%) trénovacej množiny. Preto je potrebné odhadnúť respektíve predikovať túto informáciu. Odhaduje sa teda trieda neoznačkovaných príkladov klasifikátorom naučeným na časti označkovaných údajov. Napokon sa dopredu označkované trénovacie príklady ako aj tie s odhadnutou triedou (teda rozšírená trénovacia množina) použijú na naučenie finálneho klasifikátora. Tab.15 dokumentuje výsledky experimentov, pre i $\in$ <1, 10>. Dosiahnuté výsledky ukázali, že predikcia kategórie neoznačkovaných trénovacích príkladov zvyšuje presnosť klasifikácie, čo je ilustrované názorne na obr.16 [Machová-Bednár-Mach, 2007].



Obr.16: Vplyv predikcie kategórie na presnosť klasifikácie

5 ZÁVER

V predloženej práci boli v rámci druhej kapitoly uvedené základné princípy algoritmov strojového učenia a výpočtová teória strojového učenia, algoritmy strojového učenia najčastejšie používané v oblasti spracovania dokumentov ako aj hodnotenie efektívnosti klasifikácie. Tretia kapitola bola venovaná problematike zloženej klasifikácie pomocou metód „boosting“ a „bagging“, ktorá zvyšuje efektívnosť klasifikácie textových dokumentov slabým klasifikátorom. Štvrtá kapitola sa venovala aplikácii metód strojového učenia v systémoch spracovania textových dokumentov, získaných z webu. Sústreďovala sa na vplyv váhovania slov na presnosť kategorizácie textov, generovanie kľúčových slov textového dokumentu, zhlukovanie dokumentov s kontrolovanou inicializáciou a na vplyv predikcie kategórie na presnosť klasifikácie.

Existuje veľká trieda úloh, ktoré boli riešené vo svetovom ale aj domácom merítku prístupmi prebratými zo strojového učenia. Preto je možné konštatovať, že v priebehu posledných rokov strojové učenie ako vedná disciplína dokázalo opodstatnenosť svojej existencie ako aj svoju užitočnosť. Klasifikačné metódy nachádzajú a budú nachádzať uplatnenie v dolovaní informácií a v dolovaní webu, tak z obsahu, štruktúry webu ako aj spôsobu jeho používania [Machová, 2004]. Zhlukovacie metódy sú vhodné pre aplikácie v elektronických informačných systémoch. Za zmienku stoja knižničné aplikácie, ďalej aplikácie súvisiace s návrhom a realizáciou internetového vyhľadávacieho mechanizmu. Otvorené naďalej zostávajú otázky labellingu zhlukov (stanovenie kategórie, ktorú dokumenty daného zhluku reprezentujú) a zhlukovania dokumentov s priradenými implicitnými viacerými kategóriami.

Jedným z možných smerov ďalšieho vývoja v tejto oblasti je riešenie úloh spojených s vyhľadávaním dokumentov a informácií na internete, pre ktoré je často typická nízka presnosť ale aj návratnosť. Ide o pálčivý a aktuálny problém dnešnej doby, ktorý si možno vynúti z princípu celkom nové algoritmy strojového učenia, ktoré budú schopné operovať nad tak formálne rôznorodou a rozsiahlou doménou, akou dnes internet je.

LITERATÚRA

[Bednár et al., 2005] Bednár, P., Butka, P., Paralic, J.: Java Library for Support of Text Mining and Retrieval. ZNALOSTI 2005, Stará Lesná, Vyd. Univerzity Palackého Olomouc, 2005, 162-169, ISBN 80-248-0755-6.

[Bradshaw, 1987] Bradshaw, G.: Learning about speech sounds: The NEXUS project. Proc. of the Fourth International Workshop on Machine Learning, Irvine, CA: Morgan Kaufmann, 1987, pp.1-11.

[Breiman, 1994] Breiman, L.: Bagging predictors. Technical Report 421, Department of Statistics, University of California at Berkeley, 1994.

[Breiman, 1996] Breiman, L.: Arcing the edge. Technical report 486, at UC Berkeley, 1996.

[Breiman et al, 1984] Breiman, L., Friedman, J.H., Olshen, R. A., Stone, C.J.: Classification and regression trees. Belmont, CA: Wadsworth, 1984.

[Clark-Nibblet, 1989] Clark, P., Nibblet, T.: The CN2 Induction Algorithm. Machine Learning, Vol.3, No.4, 1989, pp. 261-284.

[Dittenbach et al., 2001] Dittenbach, M., Rauber, A., Merkl, D.: Recent Advances with the Growing Hierarchical Self-Organizing Map In: Proceedings of the 3rd Workshop on Self-Organizing Maps 13-15 Jun 2001, Lincoln, England, Springer, 2001.

[Dumais, 1991] Dumais, S.: Enhancing performance in LSI retrieval. Technical Report 91/09/17, Bellcore, 1991.

[Fisher, 1987] Fisher, D. H.: Knowledge acquisitio via incremental conceptual clustering . Machine Learning, No.2, 1987, 139-172.

[Han et al., 2005] Han, K., Song, Y., Rim, H.: Probabilistic Model for Definitional Question Answering. Dept. of Computer Science& Engineering, Korea University, Seoul, Korea, 2005.

[Chakrabarti, 2003] Chakrabarti, S.: Mining the web, Discoveringknowledge form hypertext data. Morgan Kaufmann publishers, 2002, ISBN-13: 9781558607545.

[Jelínek, 2005] Jelínek, J.: Využití vazeb mezi termy pro podporu užívatele WWW. ZNALOSTI 2005, Stará Lesná, Vydavatelství Univerzity Palackého Olomouc, 2005, 218-225, ISBN 80-248-0755-6.

[Kaelbling at all, 1996] Kaelbling, L.P., Littman, M.L., Moore, A.W.: Reinforcement learning: A survey. Journal of AI Research, 4, 1996, pp. 237-285.

[Kanungo, 2002] Kanungo, T. at all.: An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Trans. Pattern Analysis and Machine Intelligence, 24*, 2002, 881-892.

[Kashef-Kamel, 2009] Kashef, R., Kamel, M.S.: Enhanced bisecting k-means clustering using intermediate cooperation. Pattern Recognition, Vol. 42, No.11, Elsevier Science Inc., New York, 2009, ISSN 0031-2569.

[Kohonen, 2001] Kohonen, T.: Self-Organizing maps. Springer-Verlang, New York, 2001, ISBN 3-540-67921-9, 3rd Edition.

[Kučera-Ježek-Hynek, 2003] M.Kučera, K.Ježek, J.Hynek: Text Categorization Using NBCI Method (in czech), ZNALOSTI 2003, Proc. pp.33-42, VŠB Technická univerzita Ostrava, ISBN80-248-0229-5

[Langley, 1996] Langley, P.: Elements of Machine Learning. Morgan Kaufmann Publishers, Inc. San Francisco, California, 1996, 419 ps.Baldonado, M., Chang, C.-C.K., Gravano, L., Paepcke, A.: The Stanford Digital Library Metadata Architecture. Int. J. Digit. Libr. 1 (1997) 108–121.

[MacQueen, 1967] 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, 281-297.

[Mach, 1997] Mach, M.: Získavanie znalostí pre znalostné systémy. Vydavateľstvo ELFA s.r.o., Košice, 1997, 104 pp.

[Machová, 1995a] Machová, K.: An Approach to Combining Inductive and Deductive Learning. Kybernetika a informatika, Slov. Spol. pre kybernetiku a informatiku, Bratislava, 1995, 68-80.

[Machová, 1995b] Machová, K.: Modifikácia vedomostí pomocou učenia založeného na vysvetľovaní. [Kandidátska dizertačná práca], FEI TU Košice, 1995, 145s.

[Machová-Janovský, 2001] Machová, K., Janovský, J.: Získavanie a modifikácia znalostí v kontexte vedomostí z oblasti prenatálnej medicíny. Lékař a Technika, Vol. 32, 2001, No. 6, Česká lékařská společnost J.E. Purkyňe, 147-151, ISSN 0301-5491.

[Machová, 2002] Machová, K.: Strojové učenie. Princípy a algoritmy. ELFA s.r.o., 2002, Košice, 117s., ISBN 80 89066 51 8.

[Machová, 2003] Machová, K.: Basic principles of the Cognitive Algorithms. Acta Elektrotechnica et Informatica, Vol. 3, No. 3, FEI TU Košice, 2003, 65-69, ISSN 1335-8243.

[Machová-Barčák-Bednár, 2006] Machová, K., Barčák, F., Bednár, P.: A Bagging Method Using Decision Trees in the Role of Base Classifiers. Acta Polytechnica Hungarica, Vol.3, No.2, 2006, Budapest Tech., Hungary, 121-132, ISSN 1785-8860.

[Machová-Bednár-Mach, 2007] Machová, K., Bednár, P., Mach, M.: Various Approaches to Web Information Processing. Computing and Informatics, Vol. 26, No. 3, 2007, Bratislava, 301-327, ISSN 1335-9150.

[Machová-Maták-Bednár, 2005] Machová, K., Maták, V., Bednár, P.: Information Extraction from the Web Pages Using Machine Learning Methods. Proc. Of the IIS – Information and Intelligent Systems – 16th International Conference, 21-23 september 2005, Varaždin, University of Zagreb, Croatia, 2005, 407-414, ISBN: 953-6071-25-8.

[Machová-Maták-Bednár, 2006] Machová, K., Maták, V., Bednár, P.: Information Retrieval from Web Pages Employing Classification and Clustering Methods. Proc. of the 5th annual conference ZNALOSTI‘06, 1-3 february 2006, Hradec Králové, Czech Republic, Publisher: VŠB-Technical University Ostrava, 2006, 191-198, ISBN: 80-248-1001-8.

[Machová-Paralič, 2003] Machová, K., Paralič, J.: Basic Principles of Cognitive Algorithms Design. Proc. of the IEEE International Conference Computational Cybernetics, Siófok, Hungary, 2003, 245-247 ISBN 963 7154 175.

[Machová, 2004] Machová, K.: An Application of Machine Learning for Internet Users. In. Emerging Solution for Future Manufacturing Systems. Springer Science+Business Media, New York, USA, 459-466, ISBN (HB) 0-387-22828-4 / e-ISBN 0-387-22829-2 (BASYS’04, Vienna, AUSTRIA, 27-29 September 2004).

[Machová-Puszta-Bednár, 2006] Machová, K., Puszta, M., Bednár, P.: An Improvement of the Classification Algorithm Results. Teaching Mathematics and Computer Science, Debrecen, Hungary, Vol.4, No.6, 2006, 131-142, ISSN 1589-7389.

[Machová at all, 2006] Machová, K., Puszta, M., Barčák, F., Bednár, P.: A Comparison of the Bagging and the Boosting Method Using the Decision Trees Classifiers. ComSIS - Computer Science and Information Systems, Vol. 3, Number 2, 2006, Novi Sad, Serbia and Montenegro, 57-72, ISSN 1820-0214.

[Machová-Szabóová, 2007] Machová, K., Szabóová, A.: Detection of Key Term Relations in Text Documents. Proc. of the 6th annual conference Znalosti 2007, Ostrava, February 21-23, 2007, Publisher: Technical University Ostrava, Czech Republic, 2007, 328-331, ISBN 978-80-248-1279-3.

[Machová-Szabóová-Bednár, 2007] Machová, K., Szabóová, A., Bednár, P.: Generation of a Set of Key Terms Characterising Text Documents. JIOS – Journal of Information and Organizational Sciences. Vol.31, NO.1, 2007, Varaždin, Croatia, 101-113, ISSN 1846-3312.

[Michalski, 1980] Michalski, R.S.: Pattern Recognition as Rule-guided Inductive Inference. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2, 1980, 349-361.

[Michalski at al, 1986] Michalski, R.S., Mozetic, I, Hong, J., Lavrac, N.: The multipurpose incremental learning system AQ15 and its testing application to three medical domains. Proc. of the Fifth National Conference on Artificial Intelligence, Philadelphia, PA: Morgan Kaufmann, 1986, pp. 1041-1045.

[Michalski-Stepp, 1983] Michalski, R. S., Stepp, R.: Learning from observation: conceptual clustering. In Michalski, R. S., Carbonell, J. G., Mitchell, T. M..: Machine Learning: An Artificial Intelligence Approach. San Mateo, CA: Morgan Kaufmann, 1983.

[Mitchell, 1997] Mitchell, T.M.: Machine Learning. The McGraw-Hill Companies, Inc. New York, 1997, 414 ps.

[Muresan-Harper, 2001] Muresan, G., Harper, D.J. (2001): Document Clustering and Language Models for System-Mediated Information Access. Proc. of the 5th European Conference on Research and Advanced Technology for Digital Libraries ECDL'01, Darmstad, September 2001, ISBN 3-540-42537-3.

[Paralič, 2002] Paralič, J.: Objavovanie znalostí v databázach. Elfa, Košice, 2002, 80s., ISBN 80-89066-60-7.

[Řezanková-Húsek-Snášel, 2007] Řezanková, H., Húsek, D., Snášel, V.: Shluková analýza dat. PBtisk, Příbram, 2007, 196s, ISBN 978-80-86946-26-9.

[Quinlan, 1990] Quinlan, J.R.: Induction of Decision trees. In. Readings in Machine Learning. Morgan Kaufmann Publishers, Inc., California, USA, 1990, 853 ps.

[Quinlan, 1993] Quinlan, J.R.: C4.5 Programes for Machine Learning. Morgan Kaufmann Publishers, San Mateo, California, 1993, ISBN 1-55860-238-0.

[Quinlan, 1996] Quinlan, J.R.: Bagging, boosting and C4.5. In Proc. of the Fourteenth National Conference on Artificial Intelligence, 1996.

[Salton-Buckley, 1988] Salton G., & Buckley C.: Term weighting approaches in automatic text retrieval. Information Processing and Management, 24(5), 1988, 513-523.

[Salton at al., 1975] Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing, Communications of the ACM, Vol.18, No.11, New York, USA, 1975, p.613-620, ISSN 0001-0782.

[Schapire-Freund, 1998] Schapire, R., Freund, Y.: Boosting the margin: A new explanation for the efectiveness of voting methods. The annals of statistics, 26(5):1651-1686, 1998.

[Schapire-Singer, 1999] Schapire, R.E., Singer, Y.: Improved Boosting Algorithms Using Confidence-rated Predictions. Machine Learning, 37(3), 1999, 297-336.

[Schapire-Singer, 2000] Schapire, R.E., Singer, Y.: BoostTexter: A Boosting – based System for Text Categorization. Machine Learning, 39(2/3), 2000, 135-168.

[Utgoff-Bradly, 1991] Utgoff, P.E., Bradly, C.F.: Incremenral induction of decision trees. Machine Learning 4, 2, 1991, pp. 161-186.

[Yiming-Pedersen, 1997] Yiming, Y., Pedersen, J.O.: A Comparative Study on Feature Selection in Text Categorization. Proceedings of ICML-97, 14th International Conference on Machine Learning, 1997.

[Zelinka, 2002] Zelinka, I.: Umělá intelligence v problémech globálni optimalizace. BEN–tech. literatura, Praha, 2002, 189, ISBN 80-7300-069-5.

Autor: Kristína Machová
Názov: Strojové učenie v systémoch spracovania informácii
Vydanie: prvé
Náklad: 50
Rozsah: 85 strán
Vydavatel: elfa s.r.o., Letná 9, Košice
Sadzba: LATEX+ gnuplot + xfig
Formát: b5
Tlac: elfa s.r.o., Letná 9, Košice
Vytlačené: 2010
ISBN