

Masarykova universita
Fakulta informatiky

Bakalářská práce

Prozkoumání systému
algoritmů strojového učení WEKA

Jakub Široký
2004

Masarykova universita
Fakulta informatiky

Bakalářská práce

Prozkoumání systému
algoritmů strojového učení WEKA

Jakub Široký
2004

Prohlašuji, že tato práce je mým původním autorským dílem, které jsem vypracoval(a) samostatně. Všechny zdroje prameny a literaturu, které jsem při vypracování používal(a) nebo z nich čerpal(a), v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

Shrnutí

Podrobný popis prostředí pro analýzu dat s použitím metod strojového učení WEKA. Pozornost je věnována jak vybraným implementovaným metodám a jejich testování, tak i uživatelskému rozhraní a snadnosti ovládání. Součástí práce je i výběr metod a dat pro výuku.

Klíčová slova

WEKA, strojové učení, učení s učitelem, Java, instance, třída, klasifikátor

Obsah

1 Úvod	6
2 Metody	7
2.1 Představení problematiky	7
2.1.1 Strojové učení	7
2.1.2 Učení s učitelem	8
2.1.3 Klasifikátor	8
2.2 Vybrané metody učení	8
2.2.1 Indukované rozhodovací stromy	8
2.2.2 Učení z příkladů	9
2.2.3 Bayesovské metody	10
2.2.4 Metody odvozování pravidel	11
2.2.5 Další metody	13
2.2.6 Metody inkrementálního učení	13
3 Program	14
3.1 Explorer	14
3.1.1 Předzpracování	15
3.1.2 Klasifikace	16
3.1.3 Další obrazovky	17
3.2 Experimenter	17
3.2.1 Příprava	17
3.2.2 Spuštění a analýza	17
3.3 KnowledgeFlow	18
3.4 CLI	19
4 Data	20
5 Testování	23
5.1 Úvodní srovnání	23
5.2 Průběžné testy a výsledky	26
5.3 Závěrem	27
6 Výuka	28
6.1 Výběr metod	28
6.2 Výběr dat	29
7 Literatura	31

1 Úvod

Program WEKA představuje prostředí pro analýzu znalostí pomocí metod strojového učení. Uživatelé nabízí sjednocený přístup k většině metod strojového učení zpracovaných ve formě samostatných programů. Převážná většina z nich je zaměřena na učení s učitelem (učení z klasifikovaných příkladů); metody učení bez učitele nebo asociativních pravidel jsou zastoupeny pouze okrajově. Vlastní práce v prostředí je usnadněna pohodlným a snadno pochopitelným grafickým uživatelským rozhraním. Grafická reprezentace najde uplatnění též při vizualizaci dat, prováděných testů a výsledků. Celé prostředí je napsané v jazyce Java, jehož vlastnosti jsou dobře využity.

Strojové učení je obor umělé inteligence, jenž se zaměřuje na pochopení a simulaci přirozeného učení. Jedním z přístupů, které se v tomto oboru uplatňují, je učení z klasifikovaných příkladů. Tomuto přístupu a výběru metod z prostředí WEKA se věnuje druhá kapitola.

Třetí kapitola popisuje uživatelská rozhraní, která odpovídají různým způsobům práce s programem. Jedná se o Explorer, Experimenter, KnowledgeFlow a CLI.

Ve čtvrté kapitole jsou ústředním tématem data použita pro testování.

Pátá kapitola se věnuje testování metod a prezentaci výsledků testů porovnávajících různá kritéria výkonnosti a efektivity.

V šesté kapitole jsou rozpracována doporučení pro výběr vzorových metod a dat pro výuku.

2 Metody

2.1 Představení problematiky

2.1.1 Strojové učení

Názvem strojové učení se souhrnně označují metody a postupy oboru umělé inteligence, které napodobují či modelují myšlenkové pochody člověka při učení. Pro tuto činnost je charakteristická nějaká úloha nebo problém, jeho případné řešení a postup vedoucí k řešení, neboli *koncept* řešení. Strojové učení se zabývá právě odvozováním či odhalováním zákonitostí vedoucích k řešení (a podstatě problému). Takto nabytých znalostí, pokud jsou dostatečně kvalitní, je následně možné využít například v expertních systémech, systémech rozpoznávání, nebo jiných oblastech, kde stroj jimi vybavený může do jisté míry (nebo úplně) zastoupit lidského odborníka.

Důležitým faktem procesu učení je zlepšování schopnosti se získanými zkušenostmi. O tento fakt se opírá i jedna z definujících vět strojového učení:

„O počítačové programu říkáme, že se učí pomocí určité zkušenosti E vzhledem k nějaké třídě úloh T a míře výkonnosti P, pokud se pro dané úlohy T jeho výkonnost měřená v P zlepšuje použitím E.“ – (Žižka, 2003).

Je dokázáno, že toto zlepšování by po nekonečně mnoha pokusech konvergovalo k vytvoření dokonalých znalostí vedoucích přesně k danému výsledku (Rao, 1978); v praxi se ve většině případů jedná o přiblížení se ideálnímu řešení.

Úspěšnost učení se hodnotí dle různých kritérií. Jedním z nich je prověření naučených znalostí proti známým, ale i neznámým datům. Sleduje se poměr správně a nesprávně určených příkladů (vyřešených úloh) podle naučeného konceptu. Jiným hlediskem pro posouzení je počet trénovacích příkladů nutných pro naučení daného konceptu (rychlost učení).

Důležitým krokem je vhodné zakódování informací nesených objekty, které reprezentují problém. Tato transformace informací do *atributů* objektů vymezuje možnosti následného vyhodnocení metodami strojového učení. Jedná se o určení množství a kvality popisů zkoumaných objektů. Lze říci, že při zadávání atributů objektů se jedná o skutečnosti zřejmé, nebo dané, při zpracování chceme získat informace skryté, získané.

Druhy hodnot atributů dále vymezují použitelnost metod pro daný problém. Některé metody jsou pracují jen s nominálními hodnotami, jiné vyžadují hodnoty numerické.

Jiným rozlišovacím znakem metod je, zda učení probíhá postupně, *inkrementálně*, nebo je *dávkové*. Inkrementální učení aktualizuje koncept po každém příkladu, učení je rychlejší, ale méně přesné. Dávkové metody využívají najednou celou množinu příkladů pro stanovení nejlepšího konceptu.

2.1.2 Učení s učitelem

Učení s učitelem, jinak také *učení z klasifikovaných příkladů*, je varianta strojového učení, kde jsou předem známa správná řešení zkoumaného problému. Výuka probíhá pomocí prezentace popisů situací a správných řešení. Úkolem žáka-algoritmu je odvodit podmínky pro platnost těchto řešení.

2.1.3 Klasifikátor

Zadáním výukové úlohy je konečná množina příkladů. Tyto příklady představují *instance* různých *tříd*. Instance jsou nejčastěji reprezentovány vektory atributů, z nichž obvykle jeden nese informaci o třídě (atribut třídy). Úkolem učícího se klasifikačního algoritmu (*klasifikátoru*) je nalezení konceptu pro rozpoznávání jednotlivých tříd.

2.2 Vybrané metody učení

Výběr metod implementovaných v prostředí WEKA s ohledem na zadání práce. V tomto zúženém přehledu jsou vynechány zejména metody pracující s číselnými třídami (regresní metody), neuronové sítě, vektorové stroje (support vector machines) a metaklasifikátory. Pro úplný seznam metod v prostředí viz (WEKA, 2004).

Pokud není stanoveno jinak, popsané metody jsou schopné pracovat s nominálními, číselnými i chybějícími atributy instancí.

2.2.1 Indukované rozhodovací stromy

Tyto metody vytvářejí řešení ve formě *n*-árních stromů, kde kořen je výchozí množina příkladů, vnitřní uzel odpovídá testu hodnoty určitého atributu a hrany představují rozdělení množiny podle výsledku testu příslušného uzlu. Koncové listy nesou výsledné rozdělení vstupní množiny. Z postupného procházení uzlů od kořene k listům je dále možné jednoduše odvodit rozhodovací pravidla.

ID3

ID3 – „Iterative Dichotomizer (version) 3“ – je základní algoritmus, který pro rozhodování, podle kterého atributu volit rozdělení, využívá entropii z teorie informace. Pro rozdělení volí takový atribut, jehož použití vede k vytvoření podmnožin s nejmenší celkovou entropií, a které tudíž obsahují nejvíce

instancí se stejnou hodnotou sledovaného atributu. Pracuje pouze s nominálními hodnotami atributů (Quinlan, 1986, Žižka, 2003).

J48

V Javě napsaná verze pokročilé metody C4.5 vycházející z původní ID3. Přidává řešení neúplně ohodnocených instancí, použití spojitých hodnot atributů, prořezávání stromů a další (Quinlan, 1993, Ingargiola, 2004).

LMT – Logistic Model Tree

Metoda odvozování jejímž základem je kombinace dvou používaných přístupů: regresních modelů a induktivních rozhodovacích stromů. Výsledkem je jediný strom nesoucí ve svých listech logistické regresní funkce. Tyto funkce dávají na rozdíl od lineárních regresních funkcí pravděpodobnostní předpověď (Landwehr, 2003). Tato metoda se ukázala být násobně (řádově ve stovkách) pomalejší než ostatní metody, a byla proto vynechána z dalšího testování.

ADTree

Alternating Decision Tree je metoda využívající při generování stromu techniku zesílení (boosting). Při každé iteraci zesílení jsou do stromu přidány tři uzly: jeden rozdělovací, jehož cílem je rozdělit množiny instancí na dvě jednoduché části, a dva rozhodovací uzly. Umístění tohoto rozdělení je dáno přezkoumáním všech rozhodovacích uzlů s ohledem na nejvyšší globální hodnocení jednoduchosti. Při hodnocení instance se počítá suma přes všechny cesty stromu, kde je možné instanci umístit, výsledek je určen jako znaménko této sumy. ADTree tady pracuje pouze s binárními třídami, lze však použít metaklasifikátor pro převod vícehodnotových tříd na binární, nejlépe stylem 1-na-1 (Holmes, 2004, WEKA, 2004).

Decision Stump

Metoda odvozování binárních „kmenů“ (stromů hloubky 1), která využívá entropii pro práci s nominálními hodnotami a regresi založenou na střední kvadratické chybě pro numerické atributy. Chybějící hodnoty vytvářejí třetí, zvláštní větev jako vlastní kategorii. Tato metoda je navržena pro práci v režimu zesílení (boosting). Základem tohoto zesílení je dělení trénovacích dat na části, natrénování více klasifikátorů a klasifikace hlasováním (Žižka, 2003).

2.2.2 Učení z příkladů

Tyto metody se učí zaznamenáváním příkladů a klasifikují podle podobnosti. Jako výsledek dotazu na neznámou instanci odhadují hodnotu jejího cílového atributu (cílové funkce, třídy) na základě cílových funkcí (tříd) různého počtu nejpodobnějších, již známých sousedů. Podobnost je dána vhodnou metrikou vzdálenosti mezi instancemi reprezentovanými n-rozměrnými vektory, například inverzí standardní euklidovské vzdálenosti. Metody této skupiny se dále liší například počtem zapamatovaných

instancí, odolností vůči šumu, způsobem výpočtu cílové funkce nebo výběrem instancí pro klasifikaci. Jejich nevýhodou je absence naučeného konceptu a potřeba poměrně rozsáhlých dat pro dobrou úroveň přesnosti. Je ale možné použít je pro základní rozdělení instančního prostoru jako podklad pro další algoritmy; například metoda LWL, obsažená v prostředí WEKA „ováhne“ část instančního prostoru, na který poté aplikuje jiný učicí algoritmus.

Metody učení z příkladů bývají označovány jako „líné“ (lazy), protože ke zpracování trénovacích dat dochází až v okamžiku dotazu na klasifikaci instance (Atkeson, 1996).

IB1

V prostředí WEKA tato metoda klasifikuje instanci pomocí jednoduché vzdálenosti nejbližšího již známého souseda. Pokud je nejbližších sousedů více, vybírá metoda prvního nalezeného (Aha, 1991, WEKA, 2004).

IBk

Rozšíření metody IB1 použitím k-nejbližších sousedů. Lze volit počet sousedů, navíc používat váhování 1/vzdálenost nebo 1-vzdálenost. Optimální počet sousedů umí metoda určit pomocí křížového ověření (cross-validation, viz kapitolu 3.1.2). Výsledná třída je spočtena jako nejčastější nebo nejtěžší třída mezi k nejbližšími sousedy. Při dávkovém učení si metoda pamatuje všechny instance, při inkrementálním stylu učení je možné určit maximální počet instancí, které si metoda bude udržovat v „oknu“ – ty nejstarší navíc budou zapomínány (Aha, 1991, WEKA, 2004).

KStar

Metoda učení z příkladů, která pro určení podobnosti využívá metriku entropické vzdálenosti. Pro dvě instance je tato metrika počítána pomocí sumy kroků všech transformací, které první instanci změní v druhou (Cleary, 1995). Tato metoda se ukázala být, podobně jako LMT, příliš neefektivní, a byla proto vynechána z dalšího testování.

LWL

Metoda Locally Weighted Learning je metaklasifikátor, který po obdržení testovací instance vybere odpovídající okolí v instančním prostoru. Na tomto okolí následně natrénuje libovolný klasifikátor a vyhodnotí testovací instanci. Doporučený klasifikátor je NaiveBayes, protože potřebuje relativně málo dat k natrénování, což umožňuje použít malé okolí instance, což zase zvyšuje šanci na výskyt nezávislých dat, jež jsou předpokladem práce tohoto algoritmu (Frank, 2003).

2.2.3 Bayesovské metody

Tyto metody využívají ke své činnosti vztahů pro výpočet podmíněné pravděpodobnosti. Trénování spočívá ve stanovení nejpravděpodobnějších hypotéz vystihujících koncept trénovacích dat. Tyto cílové pravděpodobnosti (závislé na datech) jsou dány nezávislými vstupními pravděpodobnostmi hypotéz a

dat, a dále platností dat za předpokladu hypotéz. V testovací fázi je instance klasifikována hodnotou, jenž má největší podporu mezi nejvýznamnějšími hypotézami. Výhodou těchto metod je možnost přímo využít nějaké znalosti o platnostech hypotéz a dat pro stanovení jejich vstupních pravděpodobností. Jinou výhodou je jejich vysoká přesnost, která je ale vyvážena značnou výpočetní náročností. Bayesovské metody implementované v prostředí WEKA pracují pouze s nominálními atributy a bez chybějících hodnot.

BayesNetK2

Algoritmus učení bayesovské sítě používající prohledávací gradientní algoritmus K2 omezený pevným pořadím proměnných. Protože kvalita celé sítě je určena jako suma (nebo součin) jednotlivých uzlů, je možné pro ohodnocení kvality sítě používat lokální metriky. K dispozici jsou metriky Bayes, Minimum Description Length (MDL), Akaike Information Criterion (AIK) a Entropy (Cooper, 1991, Bouckaert, 2004).

BayesNetB

Tato metoda používá gradientní algoritmus, který není omezený pevným pořadím proměnných. Volby hodnotících metrik jsou stejné jako u předešlé metody (Buntine, 1991, Bouckaert, 2004).

NaiveBayes

Metoda používající zjednodušující předpoklad nezávislosti atributů na cílovém atributu. Počet zkoumaných případů instancí se tak sníží a výpočet výrazně urychlí. Pro učení využívá frekvence kombinací hodnot v trénovacích datech. Výsledné zařazení testované instance se počítá na základě jednotlivých pravděpodobností jejích atributů pro možné hodnoty třídy a vybírá se hodnota s nejvyšší pravděpodobností (Žižka, 2003). Metoda je univerzální a poradí si s chybějícími hodnotami.

2.2.4 Metody odvozování pravidel

Tyto metody umožňují uživateli pochopit význam klasifikace tím, že vracejí sadu if-then pravidel reprezentujících znalosti odvozené z předložených dat.

ZeroR

Jednoduchý pseudoklasifikátor, který v případě nominálních atributů vrací pro libovolnou testovací instanci nejčastější ohodnocení (případně normalizované podle vah, pokud jsou váhy použity) mezi trénovacími instancemi, při práci s číselnými hodnotami vrací průměrnou hodnotu cílových atributů trénovacích dat. Využití spočívá ve stanovení základní úrovně pro posouzení výkonnosti jiných klasifikátorů (WEKA, 2004).

OneR

Metoda, která vytváří jediné pravidlo (1-rule, odpovídající stromu jediné úrovně) podle atributu, jehož použití k rozdělení množiny instancí vede k nejmenší klasifikační chybě. U numerických atributů, které diskretizuje, lze ovlivnit minimální počet intervalů, na něž se budou rozsahy hodnot dělit (Holte, 1993, WEKA, 2004).

Prism

Jednoduchá metoda odvozování rozhodovacích pravidel, která pracuje pouze s nominálními atributy, neprovádí prořezávání pravidel a neumí se vypořádat s chybějícími hodnotami (Cendrowska, 1987, WEKA, 2004).

Decision Table

Rozhodovací tabulka zpracovaná jako klasifikátor. Prostor instancí prohledává algoritmem best first pro nalezení nejvýznamnějších atributů. Hloubka prohledávání je omezena počtem po sobě jdoucích nezlepšujících se podmnožin atributů. Ohodnocení podmnožin je dáno křížovým ověřením (cross-validation). Instance, které po naučení nelze ohodnotit pomocí vzniklých pravidel, se určí buď pomocí většinového pravidla nebo pomocí nejbližšího souseda (Kohavi, 1995, Hewett, 2002, WEKA, 2004).

NNge

Hybridní metoda Non Nested Generalized Exemplars spojuje vlastnosti učení založeného na příkladech a odvozování rozhodovacích pravidel. NNge používá k ukládání informací zobecněné příklady, což jsou objekty vzniklé z instancí podobných vlastností. Toto shlukování výrazně snižuje počet objektů udržovaných v paměti. Instance, které jsou úplně obsažené v některém zobecněném příkladu, jsou zapomenuty. Instance, která nese novou informaci, vhodně rozšíří popisnou schopnost nejbližšího zobecněného příkladu stejné třídy. Pro určování vzdálenosti používá upravenou euklidovskou metriku s dynamickými změnami vah mylně klasifikujících zobecněných příkladů. Schopnosti popisu části instančního prostoru lze vyjádřit pomocí rozhodovacích pravidel. „Non nested“ v názvu metody značí, že metoda nedovoluje zanoření nebo překrývání prostorů zobecněných příkladů (Martin, 1995).

JRip

Javovská verze algoritmu RIPPER (Repeated Incremental Pruning to Produce Error Reduction), což je metoda, která pracuje ve dvou fázích. V první vzniká počáteční soubor pravidel a proběhne jejich prořezání, ve druhé dochází k jejich optimalizaci za pomoci heuristik pracujících s délkou pravidel (Cohen, 1995, Frank, 1998, WEKA, 2004).

PART

Metoda, která odvozuje pravidla opakovaným generováním částečných stromů z metody C4.5. Tyto částečné rozhodovací stromy mohou obsahovat nedefinované podstromy. Při jejich tvorbě se zároveň používá prořezávání ke zjednodušení vzniklé struktury. Po nalezení částečného stromu, který nelze dále pomocí řezu zjednodušit, je vyčteno pravidlo a strom zapomenut. Při rozhodování, který podstrom expandovat, se vybírá ten s nejmenší celkovou entropií. Instance jsou postupně procházeny technikou rozděl a panuj v tom smyslu, že po vytvoření pravidla (a vygenerování částečného stromu), se odstraní instance jím pokryté a pokračuje se ve vytváření na zbývajících datech, dokud nejsou všechny instance vyčerpané (Frank, 1998).

2.2.5 Další metody

HyperPipes

Jednoduchý klasifikátor, který vytváří prostory tříd jako vektory spojitých intervalů daných krajními hodnotami atributů instancí těchto tříd. V případě nominálních hodnot se při aktualizaci prostoru třídy hodnoty přepisují. Klasifikace spočívá v nalezení třídy, která svými intervaly nejlépe pokrývá hodnoty atributů rozpoznávané instanci. Nepracuje s číselně zadanou třídou a chybějícími atributy v testovacích datech (WEKA, 2004).

VFI

Metoda Voting Feature Intervals popisuje naučený koncept pomocí množiny intervalů vlastností. Pro všechny atributy se postupně stanoví intervaly platné pro různé třídy. Tyto intervaly mohou pokrývat instance více tříd. V intervalech odpovídajících stejným třídám narůstá počet výskytů těchto tříd. Lze využít váhování atributů dané mírou jistoty intervalu počítané pomocí entropie. Ohodnocení instance probíhá hlasováním. Nepracuje s číselným atributem třídy a ignoruje chybějící hodnoty (Demiroz, 1997, Demiroz, 2003, WEKA, 2004).

2.2.6 Metody inkrementálního učení

IB1, IBk, NaiveBayesUpdateable

Jedná se o verze již zmíněných algoritmů IB1, IBk a NaiveBayes, které pracují s postupným zpřesňováním vyučovaného konceptu. Děje se tak za pomoci ukládání a aktualizace modelů konceptů. O modelech hovoří více podkapitola 3.1.2

Jako zdroj informací o metodách posloužila převážně dokumentace k programu WEKA, dále v ní odkazované publikace autorů algoritmů a internetové zdroje. Samostatně uvedená jména značí použitou literaturu nebo publikaci autora, ve skupinách jmen je uvedena jako první autorova práce, pak použitá literatura.

3 Program

V prostředí WEKA je možné se pohybovat ve čtyřech odlišně koncipovaných grafických rozhraních: Explorer, Experimenter, KnowledgeFlow a CLI. Po spuštění se zobrazí uvítací obrazovka s nabídkou těchto uživatelských rozhraní a vyobrazením ptáčka, podle něhož je program také pojmenován. Instalaci a spuštění programu popisuje Příloha B.

3.1 Explorer

Grafické uživatelské rozhraní Explorer (Průzkumník) představuje pohodlný přístup prakticky ke všem funkcím programu potřebných k analýze a zpracování dat. Zaměřuje se na práci s jednou databází pomocí sady postupně používaných nástrojů. Tomu odpovídá rozvržení obrazovek rozhraní.

3.1.1 Předzpracování

Na první obrazovce Preprocess je možné provádět načítání dat z různých externích zdrojů, včetně internetového URL a libovolné databáze kompatibilní s Java JDBC. Následně je data možné zpracovávat pomocí *filtrů*, které jsou k dispozici. Tyto filtry slouží k úpravám tvaru dat, jednotlivých instancí a atributů. Zahrnují například doplnění chybějících hodnot, náhodné rozdělení dat na různé velké části, spojování hodnot, převod číselných hodnot na nominální, a další. Takto upravená data je možné opět uložit pomocí nativního formátu ARFF, o kterém pojednává kapitola 4.

K dalším funkcím první obrazovky patří popis načtených dat na úrovni atributů a možnost zobrazení četností jednotlivých hodnot atributů (histogram) s vyznačeným zastoupením jednotlivých tříd pomocí grafů.

3.1.2 Klasifikace

Druhá obrazovka Classify je určena k výběru klasifikátoru, jeho natrénování, otestování a zhodnocení. Na výběr je velké množství algoritmů seskupených podle přístupů ke strojovému učení. Při jejich výběru je zároveň možné nastavit jejich parametry. Ve valné většině jsou výchozí nastavení zárukou dobrých výsledků.

Natrénování probíhá s pomocí načtených dat, a to buď s celou množinou instancí nebo její částí, podle způsobu testování. Je možné testovat na *trénovacích* datech, na externích *testovacích* datech, pomocí křížového ověření nebo pomocí procentuální části trénovacích dat vyhrazené pro testování.

Na výsledky testování na trénovacích datech je možné pohlížet jako na horní odhady úspěšnosti daného klasifikátoru, pro trénování i testování jsou totiž použita stejná data.

Pro testování klasifikátoru pomocí externích dat se pro trénování použijí všechna načtená data, pro testování lze data nahrát (pouze ze souboru ARFF).

Při *křížovém ověření* program dělí načtená data na trénovací a testovací části tak, že jsou postupně všechny části použity k testování. Počet rozdělení je zvolen uživatelem a určuje poměr, v jakém se data dělí. Pro trénování se použijí vždy všechny díly s výjimkou jednoho, který bude použit pro testování. Tento postup zlepšuje odhad úspěšnosti klasifikátoru na daných datech, protože má větší šanci postihnout různé průběhy klasifikace. Pokud jsou třídy instancí nominální, je křížové rozdělení *stratifikované*, čili poměry zastoupení tříd v různých dílech odpovídají poměrům v celku.

Pro testování na části trénovacích dat je možné zvolit procentuální díl pro trénování, zbytek se použije pro testování. Jak křížové ověření, tak i procentuální dělení vybírá instance náhodně, je možné nastavit základ pro generování náhodných čísel (random seed).

Program předpokládá, že poslední atribut každé instance označuje její třídu, což je možné změnit a vybrat libovolný atribut.

Dále je možné nadefinovat *matici cen*, což je způsob, jak ohodnotit určitá klasifikační rozhodnutí penalizací a ovlivnit tak chování klasifikátoru. Tato matice se ukládá do externího souboru, který Explorer při běhu algoritmu načte. Jeho formát je uveden mezi dalšími formáty v Příloze A.

Po každém trénování vzniká *model* nesoucí naučený koncept. Tento model je možné uložit v binárním serializovaném tvaru (Java serialized object) a později znovu načíst a použít ke klasifikaci. Použije se stejný postup nahrání testovacích instancí, jako při testování na externích datech. Nabízený model vždy pochází z tréninku nad celou množinou načtených instancí a například modely z rozdělení při křížovém ověření nelze uložit. Požadovaného výsledku lze docílit pomocí filtrů.

Po natrénování klasifikátoru dochází okamžitě k jeho testování. Výsledky jsou prezentovány v textové podobě a zahrnují informace o podmínkách experimentu, popis vzniklého konceptu (pokud je to možné), dobu potřebnou k jeho vytvoření a ohodnocení úspěšnosti klasifikace. Dále je možné předvést graf chybných rozhodnutí nebo různé statistické parametry výsledků. U několika metod rozhodovacích stromů lze zobrazit vygenerovaný strom.

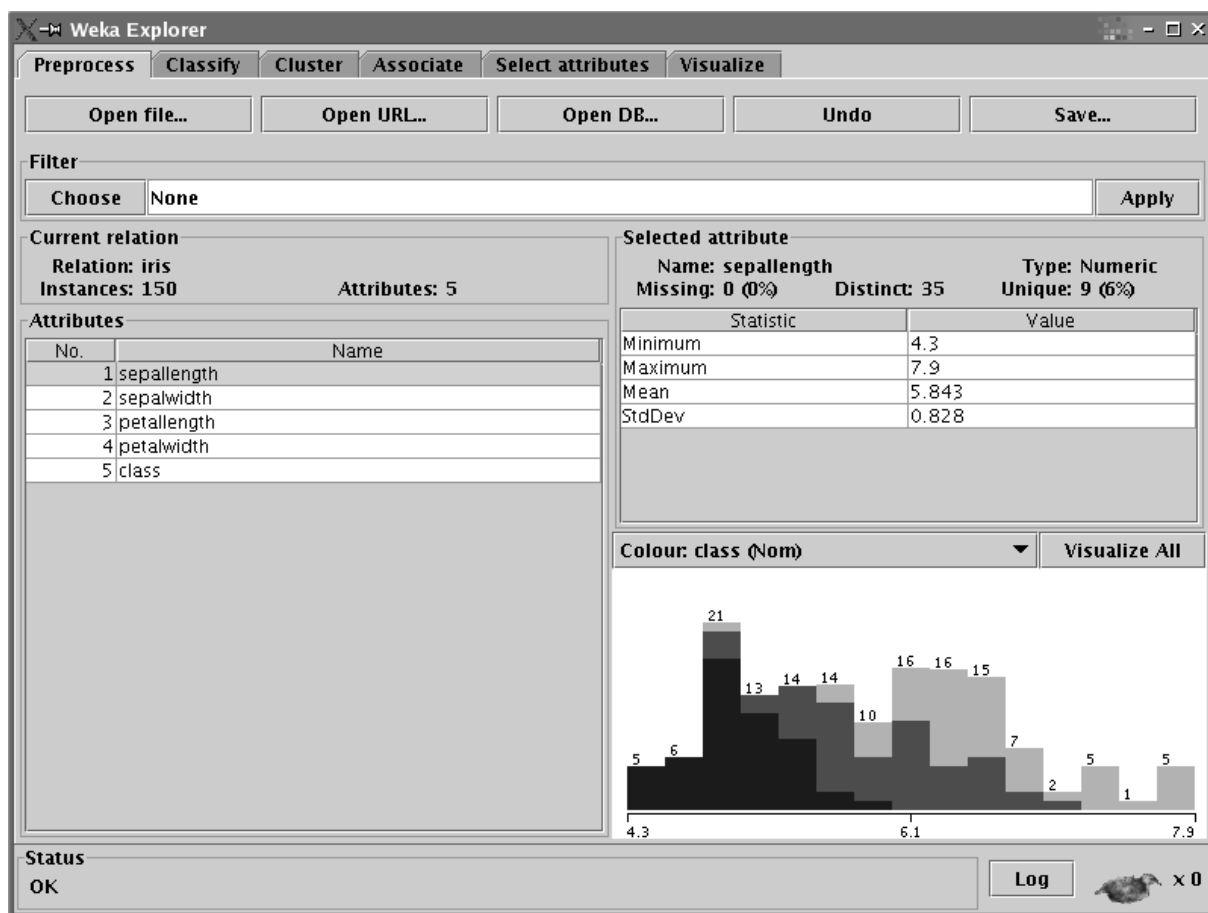
O právě probíhající činnosti program informuje pomocí jednoduché animace weky, zmíněného ptáčka, a tuto činnost si poznamenává také do souboru záznamů (log).

3.1.3 Další obrazovky

Obrazovka s názvem Cluster umožňuje práci se shlukovými metodami učení bez učitele, obrazovka Associate nabízí dvě metody asociativních pravidel.

Na obrazovce Select attributes je možné pomocí různých kritérií a různých prohledávacích technik hodnotit význam jednotlivých atributů. Tuto informaci lze využít například pro zmenšení objemu zpracovávaných dat.

Poslední obrazovka Visualize nabízí propracovaný systém grafického znázornění načtených dat. Základem je tabulka grafů představující dvourozměrné pohledy dané všemi kombinacemi dvojic atributů. Tyto grafy dále nabízejí podrobnosti až do úrovně instance. Zajímavou vlastností je možnost



Obrázek 3.1 Prostředí Explorer s načtenou databází iris

intuitivního modelování zobrazených dat pomocí výběrů instancí přímo v grafech a jejich následného exportu jako nové sestavy do souboru ARFF.

3.2 Experimenter

Grafický modul Experimenter (Experimentátor) je zaměřený na provádění rozsáhlých testovacích sestav – experimentů. Experiment je dán jako posloupnost testů provedených vybranými klasifikátory na množině datových souborů. Výsledky testování, jejichž základní struktura odpovídá výsledkům v Exploréru, je možné ukládat jako strukturovaná data a následně analyzovat pomocí statistických postupů. Zajímavá je schopnost spouštět experimenty distribuovaně po síti a dále využívat databáze k uchovávání dat. Modul je opět rozdělen do několika obrazovek podle činnosti.

3.2.1 Příprava

Obrazovka Setup je určena k vytvoření experimentu pomocí jednoduchého nebo pokročilého režimu nastavení. Jedná se o vybrání dat, klasifikátorů a způsobu testování.

V jednoduchém režimu je možné provádět stratifikované křížové ověření a procentuální rozdělení s náhodným nebo pevným rozložením dat.

V pokročilém režimu je navíc k dispozici průměrování výsledků, načítání instancí z databáze a učení na postupně se zvětšujícím objemu dat (učicí křivka). Tyto postupy je možné vhodně kombinovat, například při určování učicí křivky použít křížové ověření pro každý objem dat.

V obou režimech je možné výsledky ukládat jako soubor instancí ARFF, soubor CSV nebo výsledky odesílat do databáze. Celou konfiguraci lze uložit a později vyvolat.

3.2.2 Spuštění a analýza

Na druhé obrazovce probíhá spouštění (a případné zastavení) experimentů společně s popisem jednotlivých akcí.

Obrazovka Analyse je místem, kde dochází k hodnocení experimentů. Výsledky je možné načíst ze souboru (ARFF), z databáze nebo z proběhlého experimentu (pokud byl jako výstupní formát zvolen formát ARFF). Pro hodnocení se používá statistický test[†], který podle zadání sloupců, řádků a kritéria vygeneruje přehlednou tabulku. Při výchozím nastavení zobrazí porovnání všech klasifikátorů (sloupce) na všech datových souborech (řádky) a jako kritérium použije procentuální úspěšnost klasifikace. Je možné volit vztažný objekt pro srovnání, výchozí je první sloupec. Dále je možné zobrazit shrnutí porovnávající sloupce (klasifikátory) mezi sebou a celkové umístění podle počtu vítězství a ztrát.

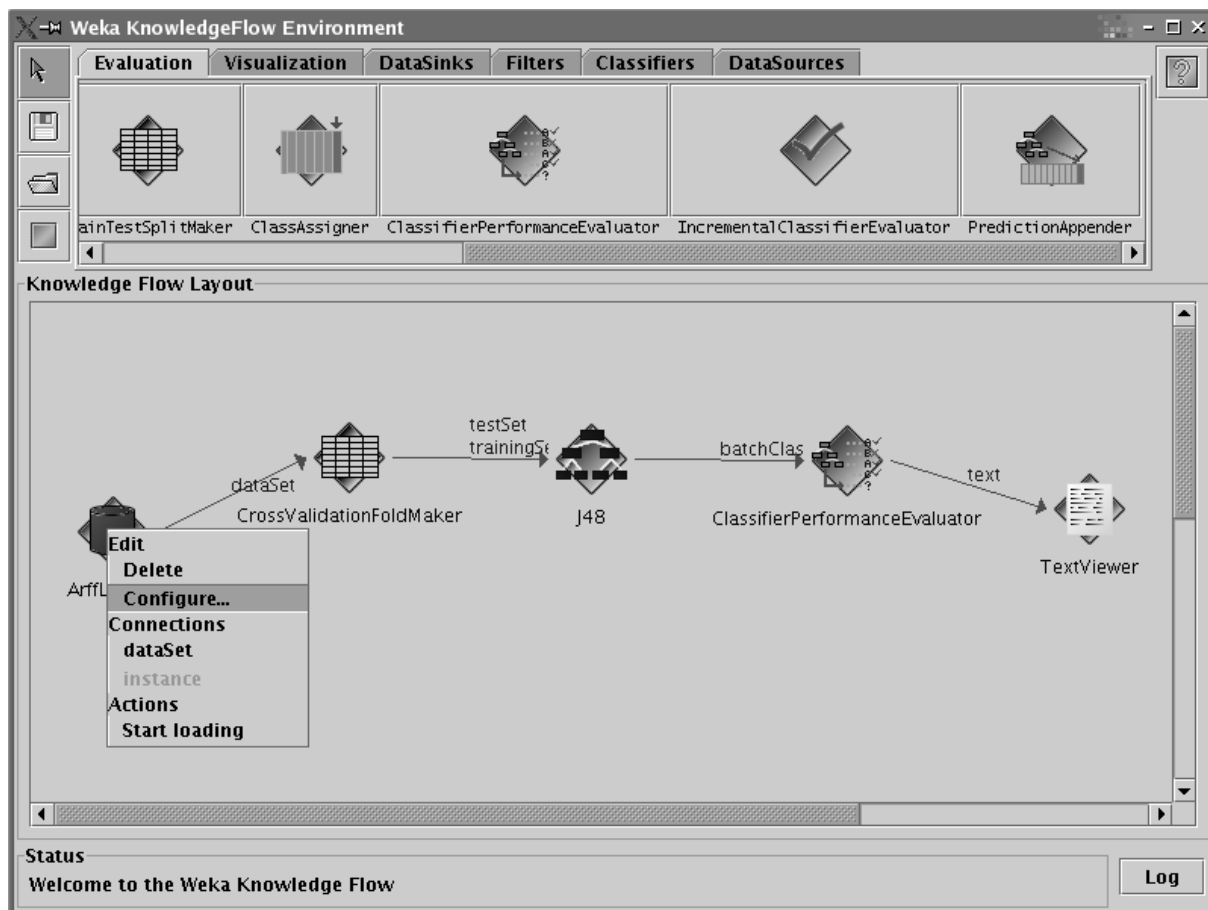
3.3 KnowledgeFlow

Třetím grafickým modulem je prostředí pro práci ve stylu diagramů datových toků. Na pracovní plochu se libovolně umísťují objekty z palety nabídek, propojují mezi sebou a vizualizují se výsledky těchto propojení. K dispozici jsou všechny klasifikátory, filtry i funkce pro načítání dat, a navíc i některé další.

Sestavení toku akcí je intuitivní a názorné. Začíná výběrem objektu pro načtení dat, přidáním objektu pro vytvoření trénovacích/testovacích dat, dále volitelně připojením objektu pro určení atributu třídy, zařazením klasifikátoru, za ním objektu pro vyhodnocení a zakončení vizualizačním objektem. Těchto toků je možné definovat více a provozovat je paralelně.

[†] jde o opravený převzorkovaný statistický t-test autorů Nadeau a Bengio, viz Nadeau, C., Bengio, Y.: Inference for the Generalization Error. Machine Learning, 2001.

Oproti zbylým rozhraním je zde možné graficky znázornit postup inkrementálního učení, řetězit filtry za sebou nebo ukázat výsledky jednotlivých rozdělení při křížovém ověření.



Obrázek 3.2 plocha KnowledgeFlow

3.4 CLI

Grafický emulátor příkazové řádky doplňuje předchozí tři rozhraní přístupem ke všem funkcím, z nichž některé jsou přístupné pouze zde. Práce zde probíhá zadáváním příkazů v podobě jmen funkcí, které se mají vykonat. Výsledky jsou prezentovány ve výstupní části okna. CLI má oproti lepšímu unixovému příkazovému řádku (shell) možnost spouštět funkce programu v rámci běžícího javovského stroje jako další vlákna (threads), čímž se šetří paměť; naopak nepříjemná je absence doplňování cest k souborům. K dispozici je několik jednoduchých systémových příkazů.

4 Data

Data používaná k testování algoritmů strojového učení mají speciální formu. Jedná se zpravidla o sadu příznaků pozorování nějakého popsatelného jevu. Může se jednat o lidskou činnost, fyzikální úkaz nebo třeba o průběh nějaké hry. Tomu odpovídá struktura těchto záznamů, zejména při učení z klasifikovaných příkladů. Pozorování jsou uchovávána a zpracována jako seznamy vektorů příznaků (atributů), přičemž poslední nese hodnocení, určení, tedy třídu daného pozorování. Pro práci s metodami se používají typické sady dat tak, aby bylo možné objektivně zkoumat vlastnosti těchto metod. V této práci jsou použita referenční data z úložiště UCI[†], data ze zemědělských výzkumů a ukázková data z instalačního balíku WEKA.

K uchování dat je v prostředí WEKA zaveden textový souborový formát ARFF, jenž srozumitelným způsobem definuje zápis instancí pomocí seznamu typovaných atributů. Zároveň umožňuje data popisovat komentáři. Popis datových formátů v prostředí WEKA je uveden v Příloze A.

Wekadata

Jedná se o tři jednoduché sestavy popsané v Tabulce 4.1.

Agrodata

Jde o 6 datových sad z oblasti zemědělského výzkumu, například o faktory ovlivňující množení jistého škůdce, nejvhodnější dobu pro sběr ovoce nebo nejlepší výběr druhu eukalyptu k vysazení. Údaje uvádí Tabulka 4.2.

UCI

Jde o výběr 34 datových sestav z různých oblastí, například měření z průmyslové výroby, bankovníctví, lékařství a jiné. Parametry dat uvádí Tabulka 4.3.

data	počet instancí	počet atributů	počet nominálních atributů	počet numerických atributů	počet tříd
contact-lenses	24	5	4	0	3
weather	14	5	2	2	2
weather.nominal	14	5	4	0	2

Tabulka 4.1 Ukázková data Wekadata

[†] UCI Machine Learning Repository je úložiště databází, doménových teorií a datových generátorů používaných k empirickému testování algoritmů strojového učení. Adresa UCI byla na jaře 2004 <http://www.ics.uci.edu/~mllearn/MLRepository.html>

data	počet instancí	počet atributů	počet nominálních atributů	počet numerických atributů	počet tříd
eucalyptus	736	20	6	13	5
grub-damage	155	9	7	1	4
pasture	36	23	2	20	3
squash-stored	52	25	4	20	3
squash-unstored	52	24	4	19	3

Tabulka 4.2 Zemědělský výzkum

data	počet instancí	počet atributů	počet nominálních atributů	počet numerických atributů	počet tříd
anneal	898	39	32	6	6
audiology	226	70	69	0	24
autos	205	26	10	15	7
balance-scale	625	5	0	4	3
breast-cancer	286	10	9	0	2
breast-w	699	10	0	9	2
colic	368	23	15	7	2
credit-a	690	16	9	6	2
credit-g	1000	21	13	7	2
diabetes	768	9	0	8	2
glass	214	10	0	9	7
heart-c	303	14	7	6	5
heart-h	294	14	7	6	5
heart-statlog	270	14	0	13	2
hepatitis	155	20	13	6	2
hypothyroid	3772	30	22	7	4
ionosphere	351	35	0	34	2
iris	150	5	0	4	3
kr-vs-kp	3196	37	36	0	2
labor	57	17	8	8	2
letter	20000	17	0	16	26
lymph	148	19	15	3	4
mushroom	8124	23	22	0	2
primary-tumor	339	18	17	0	22
segment	2310	20	0	19	7
sick	3772	30	22	7	2
sonar	208	61	0	60	2
soybean	683	36	35	0	19
splice	3190	62	61	0	3
vehicle	846	19	0	18	4
vote	435	17	16	0	2

Tabulka 4.3 34 datových sestav UCI

5 Testování

Cílem testování bylo zjistit, zda program WEKA nabízí stabilní a použitelné prostředí pro analýzu dat, dále zda používání různých rozhraní vede na stejných datech a testech ke stejným výsledkům, ověření funkce křížového ověření a průměrování experimentů.

5.1 Úvodní srovnání

Byly porovnány všechny algoritmy ve třech skupinách: univerzální, omezené na nominální atributy a na binární. Při testování nominálních atributů a binárních tříd byly přítomné i ostatní univerzální algoritmy. Tento test zároveň posloužil jako zkouška stability: na počítači s procesorem AMD s označením Athlon 2500, s operační pamětí 768MB běžel přibližně 6 hodin. Výsledky shrnuje tabulka 5.1, 5.2, 5.3, 5.4, 5.5, 5.6. Úplný přehled testů je pak v Příloze C.

Tabulka 5.1 Shrnutí testů všech klasifikátorů; udává, kolikrát byl klasifikátor ve sloupci lepší než klasifikátor v řádku (např. n byl v 11 případech lepší než a)

Analysing: Percent_correct																
Datasets: 34																
Resultsets: 15																
Confidence: 0.05 (two tailed)																
Date: 20.5.04 20:31																
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	(No. of datasets where [col] >> [row])	
-	0	0	3	4	4	4	0	0	1	3	1	2	0	0	a = trees.J48 '-C 0.25 -M 2'	
20	-	10	18	18	21	18	0	11	22	20	20	21	11	16	b = trees.DecisionStump ''	
17	1	-	14	15	16	15	0	9	17	16	18	18	10	9	c = (1)	
7	1	2	-	7	10	8	0	1	4	4	6	7	1	2	d = lazy.IB1 ''	
10	1	1	5	-	10	5	0	1	8	7	8	9	1	2	e = lazy.IBk '-K 10 -W 0'	
5	0	1	1	2	-	2	0	1	3	4	5	5	0	1	f = lazy.LWL '-W 0 -K 10 -W bayes.NaiveBayes	
--																
15	3	5	10	10	13	-	0	4	13	12	13	14	2	4	g = bayes.NaiveBayes ''	
30	30	30	28	32	31	31	-	27	30	29	30	30	17	25	h = rules.ZeroR ''	
19	3	10	16	17	21	16	1	-	21	20	19	20	8	13	i = rules.OneR '-B 6'	
6	0	0	4	5	8	5	0	0	-	7	6	6	0	1	j = rules.DecisionTable '-X 1 -S 5'	
6	1	1	5	6	7	4	0	0	2	-	1	4	0	1	k = rules.NNge '-G 5 -I 5'	
4	0	0	5	3	4	2	0	0	0	2	-	2	0	2	l = rules.JRip '-F 3 -N 2.0 -O 2 -S 1'	
2	0	0	2	4	5	5	0	0	1	3	1	-	0	0	m = rules.PART '-M 2 -C 0.25 -N 3 -Q 1'	
27	18	19	24	24	27	24	4	16	26	27	25	27	-	18	n = misc.HyperPipes ''	
19	4	6	12	12	18	14	1	4	15	17	18	19	2	-	o = misc.VFI '-B 0.6'	
(1)	meta.AdaBoostM1	'-P 100 -S 1 -I 10 -W trees.DecisionStump'														

Tabulka 5.2 Shrnutí testů binárních a univerzálních klasifikátorů

Analysing: Percent_correct																
Datasets: 16																
Resultsets: 16																
Confidence: 0.05 (two tailed)																
Date: 21.5.04 7:45																
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	(No. of datasets where [col] >> [row])
-	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	a = trees.J48 '-C 0.25 -M 2'
2	-	0	0	0	0	1	1	0	0	1	1	1	2	0	0	b = trees.ADTree '-B 10 -E -3'
3	4	-	4	3	3	4	4	0	0	5	4	3	5	0	4	c = trees.DecisionStump ''
2	1	0	-	1	2	1	2	0	0	2	1	2	2	0	0	d = (1)
5	5	0	1	-	2	4	3	0	0	2	2	3	3	0	1	e = lazy.IB1 ''
4	2	0	1	0	-	1	1	0	0	2	1	2	2	0	1	f = lazy.IBk '-K 10 -W 0'
3	3	0	1	0	0	-	0	0	0	1	2	3	3	0	1	g = lazy.LWL '-W 0 -K 10 -W
bayes.NaiveBayes																
6	5	3	5	2	4	5	-	0	3	4	3	5	5	0	2	h = bayes.NaiveBayes ''
12	12	13	13	11	14	13	13	-	12	12	12	12	12	3	9	i = rules.ZeroR ''
4	8	0	6	3	5	5	5	0	-	6	5	5	6	0	4	j = rules.OneR '-B 6'
1	0	0	0	1	0	0	0	0	0	-	1	1	2	0	1	k = rules.DecisionTable '-X 1 -S 5'
3	0	0	0	1	1	0	0	0	0	1	-	0	1	0	0	l = rules.NNge '-G 5 -I 5'
0	0	0	0	1	0	0	0	0	0	0	0	-	1	0	2	m = rules.JRip '-F 3 -N 2.0 -O 2 -S 1'
0	0	0	0	0	0	1	1	0	0	1	1	0	-	0	0	n = rules.PART '-M 2 -C 0.25 -N 3 -Q 1'
13	12	11	13	11	14	13	12	3	11	13	13	12	13	-	10	o = misc.HyperPipes ''
7	6	3	4	3	4	4	4	1	2	4	5	4	5	1	-	p = misc.VFI '-B 0.6'
(1) meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W trees.DecisionStump'																

Tabulka 5.3 Shrnutí testů nominálních a univerzálních klasifikátorů

Analysing: Percent_correct																
Datasets: 7																
Resultsets: 16																
Confidence: 0.05 (two tailed)																
Date: 20.5.04 20:36																
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	(No. of datasets where [col] >> [row])
-	3	1	2	1	2	3	2	1	0	1	1	0	2	1	1	a = trees.Id3 ''
0	-	0	0	0	0	0	1	0	0	0	0	0	0	0	0	b = trees.J48 '-C 0.25 -M 2'
5	5	-	2	5	5	5	5	0	3	5	5	5	5	3	3	c = trees.DecisionStump ''
5	5	0	-	4	4	5	3	0	3	5	5	5	5	3	2	d = (1)
1	3	1	2	-	2	2	2	0	1	1	1	1	2	0	0	e = lazy.IB1 ''
2	4	0	1	1	-	3	2	0	0	1	2	2	2	0	0	f = lazy.IBk '-K 10 -W 0'
1	1	0	0	0	0	-	0	0	0	1	0	1	1	0	0	g = lazy.LWL '-W 0 -K 10 -W
bayes.NaiveBayes																
2	3	1	3	1	3	4	-	0	2	2	3	2	3	1	1	h = bayes.NaiveBayes ''
6	6	5	5	6	6	6	6	-	6	6	6	6	6	4	5	i = rules.ZeroR ''
4	6	0	1	4	6	5	4	0	-	5	6	5	5	3	3	j = rules.OneR '-B 6'
2	3	0	0	1	0	2	3	0	0	-	0	1	2	0	0	k = rules.Prism ''
1	2	0	0	0	0	1	0	0	0	0	-	1	1	0	0	l = rules.DecisionTable '-X 1 -S 5' 26
1	2	0	0	0	0	1	1	0	0	0	0	-	0	0	0	m = rules.NNge '-G 5 -I 5'
1	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	n = rules.PART '-M 2 -C 0.25 -N 3 -Q 1'
4	5	2	2	4	4	5	4	0	3	3	4	5	5	-	2	o = misc.HyperPipes ''
4	7	1	2	3	3	5	3	0	1	4	4	5	6	1	-	p = misc.VFI '-B 0.6'
(1) meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W trees.DecisionStump'																

Tabulka 5.4 Počty vítězství a ztrát všech klasifikátorů; 1. sloupec udává vítězství-ztráty, 2. vítězství, 3. ztráty

Analysing: Percent_correct
 Datasets: 34
 Resultsets: 15
 Confidence: 0.05 (two tailed)
 Date: 20.5.04 20:32

```
>-<      >      < Resultset
165 195    30 lazy.LWL '-W 0 -K 10 -W bayes.NaiveBayes --'
165 187    22 trees.J48 '-C 0.25 -M 2'
161 184    23 rules.PART '-M 2 -C 0.25 -N 3 -Q 1'
147 171    24 rules.JRip '-F 3 -N 2.0 -O 2 -S 1'
133 171    38 rules.NNge '-G 5 -I 5'
115 163    48 rules.DecisionTable '-X 1 -S 5'
 91 159    68 lazy.IBk '-K 10 -W 0'
 87 147    60 lazy.IB1 ''
 35 153   118 bayes.NaiveBayes ''
-67  94   161 misc.VFI '-B 0.6'
-90  85   175 meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W trees.DecisionStump'
-130 74   204 rules.OneR '-B 6'
-164 62   226 trees.DecisionStump ''
-254 52   306 misc.HyperPipes ''
-394  6   400 rules.ZeroR ''
```

Tabulka 5.5 Počty vítězství a ztrát binárních a univerzálních klasifikátorů

Analysing: Percent_correct
 Datasets: 16
 Resultsets: 16
 Confidence: 0.05 (two tailed)
 Date: 20.5.04 20:50

```
>-<      >      < Resultset
 62  65     3 trees.J48 '-C 0.25 -M 2'
 58  62     4 rules.PART '-M 2 -C 0.25 -N 3 -Q 1'
 49  53     4 rules.JRip '-F 3 -N 2.0 -O 2 -S 1'
 49  58     9 trees.ADTree '-B 10 -E -3'
 48  55     7 rules.DecisionTable '-X 1 -S 5'
 44  51     7 rules.NNge '-G 5 -I 5'
 35  52    17 lazy.LWL '-W 0 -K 10 -W bayes.NaiveBayes --'
 32  49    17 lazy.IBk '-K 10 -W 0'
 32  48    16 meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W trees.DecisionStump'
  7  38    31 lazy.IB1 ''
 -5  47    52 bayes.NaiveBayes ''
-16  30    46 trees.DecisionStump ''
-22  35    57 misc.VFI '-B 0.6'
-34  28    62 rules.OneR '-B 6'
-169  4   173 rules.ZeroR ''
-170  4   174 misc.HyperPipes ''
```

Tabulka 5.6 Počty vítězství a ztrát nominálních a univerzálních klasifikátorů

```
Analysing: Percent_correct
Datasets: 7
Resultsets: 16
Confidence: 0.05 (two tailed)
Date: 20.5.04 20:36

>-< > < Resultset
54 55 1 trees.J48 '-C 0.25 -M 2'
44 45 1 rules.PART '-M 2 -C 0.25 -N 3 -Q 1'
42 47 5 lazy.LWL '-W 0 -K 10 -W bayes.NaiveBayes --'
34 39 5 rules.NNge '-G 5 -I 5'
31 37 6 rules.DecisionTable '-X 1 -S 5'
20 34 14 rules.Prism ''
18 39 21 trees.Id3 ''
15 35 20 lazy.IBk '-K 10 -W 0'
11 30 19 lazy.IB1 ''
5 36 31 bayes.NaiveBayes ''
-32 17 49 misc.VFI '-B 0.6'
-34 20 54 meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W trees.DecisionStump'
-36 16 52 misc.HyperPipes ''
-38 19 57 rules.OneR '-B 6'
-50 11 61 trees.DecisionStump ''
-84 1 85 rules.ZeroR ''
```

5.2 Průběžné testy a výsledky

Stabilita:

Pro základní práci je stabilita prostředí dobrá, existují však chyby v implementacích některých funkcí, například při použití více úrovní vnoření metaklasifikátorů. Záleží též na prostředí, ve kterém je program provozován: v operačních systémech Windows a Linux s použitím Java2 verze 1.4.2 byla stabilita vyšší než v systému FreeBSD se stejnou verzí Javy, kde docházelo k chybám při výpočtech a návratu NaN hodnoty.

Nároky na vybavení:

Nejsou přemrštěné, program WEKA poběží všude tam, kde bude k dispozici prostředí Java verze 1.4.2 a vyšší. Je však smysluplné vybavit počítače dostatkem paměti, při testování na UCI datech nebyly výjimkou objemy dat v pracovní paměti přesahující 500MB.

Křížové ověření automatické a zabudované:

Oba přístupy vedou ke stejným výsledkům, lze tedy očekávat, že funkce Cross-Validate, která je součástí každé klasifikace (pokud není zadán testovací soubor), funguje správně.

Náhodné rozdělení:

Stejně jako při křížové ověření dává zabudovaná funkce spolehlivé výsledky.

Použitelnost pro výzkum a výuku:

Velmi názorné prostředí KnowledgeFlow nabízí rychlou cestu k pochopení fungování programu a strojového učení vůbec. V zatím poslední verzi 3.4.1 má některé nedostatky, které se ale dají překonat.

Prostředí Explorer je velmi mocný nástroj, kde lze názorně provádět analýzu a úpravu dat. Možnosti vizualizace v tomto prostředí jsou opravdu široké.

Prostředí Experimenter je ideálním pomocníkem při testování nových metod, zkoumání velkého množství dat a dávkových úloh. Lze jej využít na zjištění optimálních hodnot parametrů klasifikátorů. Implicitní hodnoty parametrů klasifikátorů v prostředí WEKA odpovídají parametrům vybraným pomocí automatického testování kombinací parametrů za pomoci křížového ověření.

Nedostatky:

Zadávání parametrů klasifikátorů při experimentu je poznamenáno drobnými implementačními chybami dialogů.

Při používání KnowledgeFlow prostředí zůstávají v paměti zbytky předchozího diagramu, je třeba restartovat celý program; dále se při načtení uložené sestavy ztratí informace o datovém souboru – toto prostředí je ale ve verzi 3.4.1 nové a je pravděpodobné, že chyby budou v příštích verzích opraveny.

Některé kombinace klasifikátorů a metaklasifikátorů nefungují (PART+CVPParameterSelection), také některé nativní parametry klasifikátorů (AdaBoost).

Při testování na optimální parametry (pomocí CVPParameterSelection, nebo bez něj) docházelo při použití prostředí Experimenter k selhání BayesNetB a BayesNetK2.

5.3 Závěrem

Celkově lze prostředí WEKA doporučit s tím, že výborně umí svou základní práci, tedy analyzovat a zpracovávat data za použití nejrozumnějších metod strojového učení. Pokud však jde o různé optimalizační techniky, ladění klasifikačních algoritmů pomocí vestavěných podpůrných prostředků a obecně testování metod, je třeba vždy ověřit chování systému na cílové oblasti použití, třeba s menším objemem dat.

Další pozornost by si určitě zasloužila schopnost programu pracovat s externí databází a možnost distribuování experimentů po síti.

6 Výuka

Pro výuku lze doporučit takové metody a taková data, která umožní pochopit podstatu různých přístupů strojového učení (zde s učitelem). K tomu je potřeba, aby objem dat a složitost algoritmů umožňovaly ověření pomocí ručního výpočtu podle principů jejich práce. Složitější metody a data je možné uvést s tím, že váha jejich využití bude přenesena na zkoumání jejich chování za různých podmínek, sestavování rozsáhlejších experimentů a dále na získání představy o postupech používaných k hodnocení metod strojového učení podle různých měřítek.

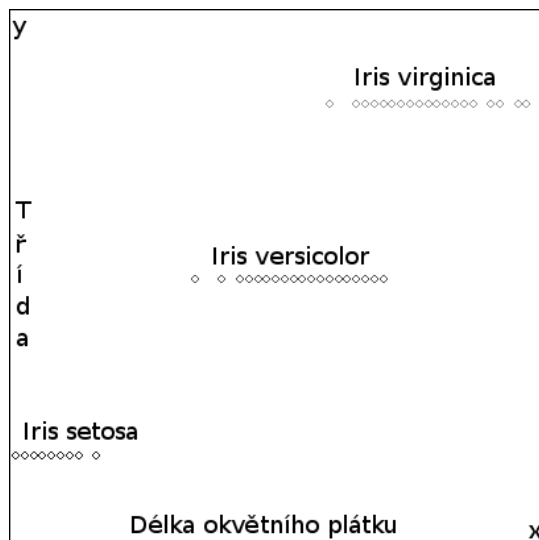
6.1 Výběr metod

- **Základní metody**, které je možné ověřit výpočtem:
 - odvozený rozhodovací strom ID3
 - algoritmy učení na příkladech IB1 a IBk, které používají nejbližší sousedy
 - metoda NaiveBayes pracující s podmíněnou pravděpodobností
 - ZeroR a OneR pro stanovení spodního odhadu výkonu
 - HyperPipes rozdělující instanční prostor podle vektorů intervalů hodnot atributů.
- **Složitější metody**:
 - DecisionStump ve spojení s vhodným algoritmem pro zesílení (LogitBoost, AdaBoost)
 - J48 jako rozšíření ID3
 - přesná síť BayesNetK2 nebo BayesNetB
 - zajímavá hybridní metoda NNge zobecněných příkladů spojující učení z příkladů s odvozováním pravidel
 - metoda DecisionTable
- **Inkrementální metody** (KnowledgeFlow: graf Chart – křivka učení v reálném čase):
 - verze IB1
 - verze IBk
 - NaiveBayesUpdateable

Obecně metody rozhodovacích stromů a pravidel produkují při použití s nominálními atributy dobře čitelný výstup.

6.2 Výběr dat

Následující soubory dat představují různorodý materiál pro použití s různými metodami strojového učení. Příklady zahrnují jak sestavy pouze s nominálními nebo pouze číselnými atributy, tak smíšené, binární nebo neúplné vzorky.



Obrázek 6.1 Rozložení tříd v závislosti na délce okvětního plátku (atribut petal length)

iris představuje asi nejznámější soubor testovacích dat o třech druzích rostliny iris, která obsahuje 3 třídy rostlin po 50 instancích, kde první je od druhých dvou lineárně separabilní, zatímco dvě zbývající mezi sebou nikoliv, jak na příkladu atributu ukazuje Obrázek 6.1.

letter databáze 20 000 instancí popisujících velká písmena anglické abecedy pro rozpoznávání. obsahuje 16 celočíselných atributů s hodnotami 0-15 a třídu odpovídající písmenu.

credit-a 690 instancí o schválení kreditních karet obsahující dobrou směs 15 atributů: spojitých, nominálních s velkými a malými rozsahy hodnot.

credit-g obsahuje 1 000 instancí spojených s výdejem německých kreditních karet. Pro testování je třeba zavést matici cen, což je způsob, jak zvýhodnit nebo penalizovat určitá rozhodnutí. Je dostupná v rozhraní Explorer z externího souboru, jehož formát popisuje Příloha A.

ionosphere obsahuje 351 instancí popisujících radarové odrazy při výzkumu ionosféry na přítomnost volných elektronů. Instance jsou zapsány pomocí 34 reálných atributů a binární třídy.

kr-vs-kp 3 196 popisů šachovnice při zakončení partie. Přibližně polovina instancí je ohodnocena Bílý může vyhrát, zbytek Bílý nemůže vyhrát. Obsahuje 35 binárních a 2 ternární atributy.

mushroom 8 124 vzorků jedlých a jedovatých amerických hub. Binární třída je rozdělena mezi instancemi přibližně napůl, všechny atributy jsou nominální. Chybí 2480 hodnot atributu 11.

zoo Jednoduchá databáze obsahující 101 instancí o 17 binárních attributech s třídou nabývající 7 hodnot.

Ukázkové databáze weather, weather.nominal a contact-lenses jsou dostatečně jednoduché pro ověření vlastními výpočty.

7 Literatura

- Aha, D., Kibler, D., Albert, M. K.: Instance-based learning algorithms. *Machine Learning*, 6, 1991, s. 37-66.
- Atkeson, C. G., Moore, A. W., Schaal, S.: Locally Weighted Learning. *Artificial Intelligence Review*, 11, 1997, 11-73. URL: <ftp://ftp.cc.gatech.edu/pub/people/cga/air1.html> (květen 2004).
- Bouckaert, R. R.: Bayesian Network Classifiers in Weka, 2004.
URL: http://www.cs.waikato.ac.nz/~remco/weka_bn (květen 2004).
- Buntine, W.: Theory refinement on Bayesian networks. In: *UAI '91: Proceedings of the Seventh Annual Conference on Uncertainty in Artificial Intelligence*, 7, Los Angeles, 1991, s. 52-60.
- Cendrowska, J.: PRISM: An algorithm for inducing modular rules. *International Journal of Man-Machine Studies*, 27, 1987, 4, s. 349-370.
- Cleary, J. G., Trigg, L. E.: K*: An instance-based learner using an entropic distance measure. In: *Machine Learning: Proceedings of the Twelvth International Conference*, 12, Tahoe City, 1995, s. 108-114. URL: <http://www.cs.waikato.ac.nz/~ml/publications/1995/Cleary95-KStar.pdf> (květen 2004).
- Cohen, W.: Fast effective rule induction. In: *Proceedings of Twelfth International Conference on Machine Learning (ICML-95)*, 12, 1995, s. 115-123.
- Cooper, G., Herskovits, E.: A Bayesian Method for Constructing Bayesian Belief Networks from Databases. In: *UAI '91: Proceedings of the Seventh Annual Conference on Uncertainty in Artificial Intelligence*, 7, Los Angeles, 1991, s. 86-94.
- Demiröz, G., Güvenir, H. A.: Classification by Voting Feature Intervals. [výzkumná práce], Department of Computer Engineering, Bilkent University, 1997.
URL: <http://www.cs.bilkent.edu.tr/tech-reports/1997/BU-CEIS-9708.ps.z> (květen 2004).
- Frank, E., Hall, M., Pfahringer, B. Locally Weighted Naive Bayes. [výzkumná práce], Department of Computer Science, University of Waikato, Hamilton, 2003.
URL: <http://www.cs.waikato.ac.nz/~mhall/FrankHallPfahringUAI2003.ps> (květen 2004).
- Frank, E., Witten, I. A.: Generating Accurate Rule Sets Without Global Optimization. In: *Machine Learning: Proceedings of the Fifteenth International Conference*, 15, San Francisco, 1998.
URL: <http://www.cs.waikato.ac.nz/~ml/publications/1998/Frank-Witten-Generating-Rules.ps> (květen 2004).
- Hewett, R., and J. Leuchner: The Power of Second-Order Decision Tables. In: *Proceedings of the second SIAM International Conference on Data Mining (SDM'2002)*, 2002.
URL: <http://www.siam.org/meetings/sdm02/proceedings/contents.htm> (květen 2004).
- Holmes, G., Pfahringer, B., Kirkby, R., Frank, E., and Hall, M.: Multiclass Alternating Decision Trees. In: *Proceedings of the European Conference on Machine Learning(2002)*. Helsinki, Springer-Verlag, 2002. URL: <http://www.cs.waikato.ac.nz/~bernhard/papers/ecml2002.pdf> (květen 2004).

- Holte, R., C.: Very simple classification rules perform well on most commonly used datasets. Machine Learning, 11, 1993,1, s. 63-90. URL: http://www.cs.pdx.edu/~timm/dm/simple_rules.pdf (květen 2004).
- Ikizler, N., Güvenir, H. A.: Maximizing Benefit of Classifications Using Feature Intervals. [výzkumná práce], Department of Computer Engineering, Bilkent University, 2003.
URL: <http://www.cs.bilkent.edu.tr/tech-reports/2003/BU-CE-0301.pdf> (květen 2004).
- Ingargiola, G.: Building Classification Models: ID3 and C4.5 .
URL: <http://www.cis.temple.edu/~ingargio/cis587/readings/id3-c45.html>, (květen 2004).
- Kohavi, R.: The Power of Decision Tables. In: Proceedings of European Conference on Machine Learning, Lecture Notes in Artificial Intelligence 914. Berlin, 1995, s. 174-189.
- Landwehr, N., Hall, M., Frank, E.: Logistic Model Trees. In: Proceedings of the European Conference on Machine Learning(2003), 14, 2003, s. 241-252.
URL: <http://www.cs.waikato.ac.nz/~mhall/LandwehrHallFrankECML2003.ps> (květen 2004).
- Martin, B.: Instance-Based learning: Nearest Neighbor With Generalization. [diplomová práce], University of Waikato, Hamilton.
URL: <http://www.cs.waikato.ac.nz/~ml/publications/1995/Martin95-Thesis.pdf> (květen 2004).
- Mařík, V., Štěpánová, O., Lažanský, J.: Umělá inteligence 1, 2, 3, 4. Praha, Academia, 1993, 1997, 2001, 2002.
- Quinlan, R.: Induction of decision trees. Machine Learning, 1, 1986, 1, s. 81-106.
- Quinlan, R.: C4.5: Programs for Machine Learning. San Mateo, Morgan Kaufmann Publishers, 1993.
- Rao, R. C.: Lineární metody statistické indukce a jejich aplikace. Academia, Praha, 1978.
- WEKA, 2004 - dokumentace javadoc k programu WEKA verze 3.4.1.
URL: http://www.cs.waikato.ac.nz/~ml/weka/doc_gui/index.html (květen 2004).
- Wilson, W.: Induction of Decision Trees, 2003.
URL: <http://www.cse.unsw.edu.au/~billw/cs9414/notes/ml/06prop/id3/id3.html> (květen 2004).
- Witten, I. H., Frank, E.: Data Mining, Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann Publishers, 2000, 8. kapitola.
- Žižka, J.: Úvod do strojového učení. [skript], podzim 2003, Fakulta informatiky MU, Brno.

Příloha A – formáty souborů

ARFF

Jedná se o nativní textový formát prostředí pro načítání a ukládání datových sad a výsledků z experimentů.

```
%1. Title: Johns Hopkins University Ionosphere database
.
.
.
.                                % popis

@relation ionosphere              % název datové sestavy

@attribute a01 real                % definice spojitého atributu
@attribute a02 integer            % definice diskrétního atributu
@attribute a03 numeric            % definice číselného atributu obecně,
                                integer je brán jako real
@attribute a04 {A,B,C}            % definice nominálního atributu
@attribute a05 string              % definice textového atributu - popisy apod.
.
.
.
.
.

@attribute class {b, g}           % definice třídy, nominální hodnota

@data                             % začátek dat
0.0609,2,1.0,C,'objekt 1',b      % první instance třídy b
.
.
.
```

CSV

Textový formát s údaji oddělenými čárkami nebo jinými znaky. Volitelně může první řádek obsahovat popisy. V prostředí WEKA lze z tohoto formátu načítat do prostředí Explorer a KnowledgeFlow. V modulu Experimenter lze ukládat výsledky proběhlých testů.

C45

Textový formát z produktu C4.5 pro strojové učení. Lze jej načítat do prostředí Explorer a KnowledgeFlow.

EXP

Formát používaný pro ukládání nastavení experimentů. Jde o Java serialized object.

MODEL

Opět Java serialized object, určený pro ukládání konceptů vytvořených klasifikátory. Struktura se liší podle původce.

ZIP

Archivní formát lze použít pro ukládání textových mezivýsledků (jednotlivá rozdělení při křížovém ověření, atd.) při běhu experimentu.

CENOVÁ MATICE

Slouží pro ovlivnění chování klasifikátoru pomocí penalizací. Výchozí penalizace každého rozhodnutí je 1. Jedná se o řádky o třech položkách, první značí skutečnou třídu, druhá chybné rozhodnutí a třetí je cena za toto rozhodnutí.

```
A B 10      % znamená, že pokud třída bude A, a klasifikátor rozhodne B,  
             bude penalizován  
B A 1       % naopak nikoliv
```


Příloha B – instalace a spuštění programu

- Pro práci s programem WEKA je potřeba Java 2 1.4.2 a vyšší.
- Instalace spočívá v rozbalení archivu s programem do zvoleného adresáře.
- Pro správnou funkci programu je třeba nastavit proměnnou WEKAHOME na adresář obsahující soubor weka.jar.
- V případě, že je potřeba využívat části programu přímo (jsou to Java třídy), je nutné přidat do proměnné CLASSPATH také soubor weka.jar včetně cesty.
- Program se spouští pomocí java virtuálního stroje příkazem:

```
java -jar weka.jar
```

nebo přímo např. Explorer:

```
java -cp weka.jar weka.gui.explorer.Explorer
```

Příloha C – výsledky testů.

data: UCI všech 34 vzorků, klasifikátory: univerzální, test: 10-ti násobné křížové ověření

Analysing: Percent_correct
Datasets: 34
Resultsets: 15
Confidence: 0.05 (two tailed)
Date: 20.5.04 20:22

Dataset	(1) trees.J48	(2) trees.	(3) meta.A	(4) lazy.I	(5) lazy.I	(6) lazy.L	(7) bayes.	(8) rules.	(9) rules.	(10) rules	(11) rules	(12) rules	(13) rules	(14) misc.	(15) misc.
anneal	(10) 98.44	77.16 *	83.63 *	99.11	95.88	98	86.3 *	76.17 *	83.63 *	98.89	99	98.33	98.22	98	71.6 *
audiology	(10) 77.87	46.46 *	46.46 *	75.22	56.11 *	78.64	73.42	25.2 *	46.46 *	75.16	71.17	73.93	78.3	66.34 *	52.94 *
autos	(10) 81.88	44.86 *	44.86 *	75.93	59.1 *	77.05	56.12 *	32.67 *	61.88 *	78.02	79.9	73.1 *	77.45	64.74 *	60.02 *
balance-scale	(10) 76.65	55.06 *	72.31	79.03	90.08 v	87.68 v	90.39 v	45.76 *	56.33 *	74.56	81.93 v	80.8	83.54 v	46.08 *	71.48
breast-cancer	(10) 75.54	68.55	70.28	65.74 *	73.09	72.11	71.7	70.3	65.74 *	72.75	65.1 *	70.95	71.33	69.95	67.14 *
wisconsin-breast-cancer	(10) 94.56	92.42	94.85	95.28	96.42	96.42	95.99	65.52 *	92.71	95.42	95.99	95.42	93.85	88.56 *	95.7
horse-colic	(10) 85.3	81.52	81.26	81.27	83.14 *	83.96	78 *	63.05 *	81.52	81.23	80.45	84.22	84.77	61.97 *	78.54 *
credit-rating	(10) 86.09	85.51	84.64	81.16	85.94	85.07	77.68 *	55.51 *	85.51	85.07	82.61	85.8	85.36	44.64 *	84.64
german_credit	(10) 70.5	70	69.5	72	74	73.1	75.4 v	70	66.9	72.2	70.5	71.7	70.2	69.9	70.9
pima_diabetes	(10) 73.83	71.87	74.35	70.17	71.1	73.44	76.31	65.11 *	72.26	73.31	73.97	76.04	75.27	65.49 *	63.94 *
Glass	(10) 66.75	44.91 *	44.91 *	70.5	66.39	71	48.59 *	35.52 *	58.46	69.11	70	68.66	68.14	51.41 *	54.59 *
cleveland-14-heart-diseas	(10) 77.52	71.55	82.11	76.22	82.46	82.13	83.47	54.45 *	71.55	76.88	80.86	81.45	79.86	55.44 *	79.83
hungarian-14-heart-diseas	(10) 81.07	79.97	77.95	76.83	83.39	81.95	83.7	63.95 *	78.63	78.6	79.64	78.95	81.02	64.28 *	83.01
heart-statlog	(10) 76.67	72.59	80	75.19	81.48	82.22	83.7	55.56 *	71.11	82.96 v	78.15	78.89	73.33	57.04 *	80
hepatitis	(10) 83.79	77.5	82.54	80.63	82.63	83.79	84.46	79.38	81.25	81.04	84.42	78	84.46	64.96 *	85.08
hypothyroid	(10) 99.58	95.39 *	93.21 *	91.52 *	93.24 *	94.46 *	95.28 *	92.29 *	96.24 *	99.39	98.7 *	99.34	99.42	93.29 *	92.34 *
ionosphere	(10) 91.46	82.62 *	90.9	86.33 *	84.89	82.9 *	82.62 *	64.1 *	80.92 *	89.46	90.04	89.75	91.75	35.9 *	94.32
iris	(10) 96	66.67 *	95.33	95.33	96	95.33	96	33.33 *	94	92.67	96	94.67	94	90.67	96
kr-vs-kp	(10) 99.44	66.05 *	93.84 *	89.96 *	95.06 *	97.56 *	87.89 *	52.22 *	66.46 *	97.65 *	98.53 *	99.19	99.06	54.1 *	88.2 *
labor	(10) 73.67	80	87.33	82.67	91.67	91.67	90	64.67 *	75.33	77	77.33	77	78.67	86	84.67
letter	(10) 87.98	7.09 *	7.09 *	96 v	94.67 v	96.77 v	64.11 *	4.06 *	17.24 *	71.54 *	91.43 v	86.28 *	88.76	22.26 *	61.15 *
lymphography	(10) 76.95	75.48	74.14	80.9	80.9	85.76	83.05	54.76 *	74.81	72.95	78.19	77.76	76.24	58.14 *	78.24
mushroom	(10) 100	88.68 *	96.2 *	100	99.94	100	95.83 *	51.8 *	98.52 *	100	100	100	100	99.77	99.9
primary-tumor	(10) 39.8	28.89 *	28.89 *	33.59	47.46 v	40.1	50.13 v	24.79 *	27.42 *	40.69	40.7	39.22	40.7	24.79 *	30.68 *
segment	(10) 96.93	28.57 *	28.57 *	97.14	94.33 *	96.88	80.22 *	14.29 *	64.55 *	91.86 *	96.28	95.15 *	96.23	75.45 *	77.36 *
sick	(10) 98.81	96.55 *	97.19 *	96.18 *	96.16 *	96.66 *	92.6 *	93.88 *	96.34 *	97.77	96.9 *	98.22	98.62	93.85 *	65.85 *
sonar	(10) 71.17	73.05	71.67	86.57 v	75.98	80.83	67.88	53.38 *	62.5	74.5	72.14	73.07	80.31	59.21 *	57.79 *
soybean	(10) 91.51	27.96 *	27.96 *	89.89	87.7	92.09	92.96	13.47 *	39.96 *	86.97 *	91.8	92.52	91.94	86.09 *	86.66
splice	(10) 94.08	62.38 *	86.74 *	75.92 *	83.26 *	86.65 *	95.3	51.88 *	24.36 *	93.1	49.47 *	94.45	92.73 *	41.79 *	88.43 *
vehicle	(10) 72.47	39.95 *	39.95 *	69.86	70.22	73.76	44.8 *	25.65 *	50.72 *	65.01 *	61.24 *	68.56	71.51	32.64 *	53.9 *
vote	(10) 96.33	95.64	95.41	92.42 *	92.89 *	93.36	90.14 *	61.38 *	95.64	95.18	96.09	95.41	94.71	61.38 *	91.74 *
vowel	(10) 81.52	17.37 *	17.37 *	99.29 v	60.1 *	96.16 v	63.74 *	9.09 *	32.53 *	71.62 *	87.47	69.09 *	76.67 *	36.87 *	57.58 *
waveform	(10) 75.08	56.76 *	66.64 *	73.62	80.6 v	80.5 v	80 v	33.84 *	54.02 *	73.8	77.86 v	79.2 v	77.42 v	46.14 *	56.16 *
zoo	(10) 92.18	60.45 *	60.45 *	97.09	88.18	95.18	95.09	40.64 *	42.64 *	91.18	95.18	87.27	92.18	94.18	94.09
(v/ /*) (0/14/20) (0/17/17) (3/24/7) (4/20/10) (4/25/5) (4/15/15) (0/4/30) (0/15/19) (1/27/6) (3/25/6) (1/29/4) (2/30/2) (0/7/27) (0/15/19)															

Skipped:

Key:

(1) trees.J48 '-C 0.25 -M 2'
(2) trees.DecisionStump ''
(3) meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W trees.DecisionStump'
(4) lazy.IB1 ''
(5) lazy.IBk '-K 10 -W 0'
(6) lazy.LWL '-W 0 -K 10 -W bayes.NaiveBayes --'
(7) bayes.NaiveBayes ''
(8) rules.ZeroR ''

(9) rules.OneR '-B 6'
(10) rules.DecisionTable '-X 1 -S 5'
(11) rules.NNge '-G 5 -I 5'
(12) rules.JRip '-F 3 -N 2.0 -O 2 -S 1'
(13) rules.PART '-M 2 -C 0.25 -N 3 -Q 1'
(14) misc.HyperPipes ''
(15) misc.VFI '-B 0.6'

data: UCI všech 34 vzorků, klasifikátory: univerzální, test: 10-ti násobné křížové ověření, průměrné hodnoty

Analysing: Percent_correct
Datasets: 1
Resultsets: 15
Confidence: 0.05 (two tailed)
Date: 20.5.04 20:23

Dataset	(1) trees.J4	(2) trees	(3) meta.	(4) lazy.	(5) lazy.	(6) lazy.	(7) bayes	(8) rules	(9) rules	(10) rule	(11) rule	(12) rule	(13) rule	(14) misc	(15) misc	
1	(340)	83.57	64.4 *	69.07	82.6	82.19	85.39	79.79	49.93 *	66.71 *	81.69	82.03	82.72	83.71	63.57 *	75.13
	(v/ /*)	(0/0/1)	(0/1/0)	(0/1/0)	(0/1/0)	(0/1/0)	(0/1/0)	(0/0/1)	(0/0/1)	(0/1/0)	(0/1/0)	(0/1/0)	(0/1/0)	(0/0/1)	(0/1/0)	

Skipped:

Key:

(1) trees.J48 '-C 0.25 -M 2'
(2) trees.DecisionStump ''
(3) meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W trees.DecisionStump'
(4) lazy.IB1 ''
(5) lazy.IBk '-K 10 -W 0'
(6) lazy.LWL '-W 0 -K 10 -W bayes.NaiveBayes --'
(7) bayes.NaiveBayes ''
(8) rules.ZeroR ''
(9) rules.OneR '-B 6'
(10) rules.DecisionTable '-X 1 -S 5'
(11) rules.NNge '-G 5 -I 5'
(12) rules.JRip '-F 3 -N 2.0 -O 2 -S 1'
(13) rules.PART '-M 2 -C 0.25 -N 3 -Q 1'
(14) misc.HyperPipes ''
(15) misc.VFI '-B 0.6'

data: UCI binární třídy, klasifikátory: ADTree a univerzální, test: 10-ti násobné křížové ověření

Analysing: Percent_correct
Datasets: 16
Resultsets: 16
Confidence: 0.05 (two tailed)
Date: 20.5.04 20:24

Dataset	(1) trees.J4	(2) trees	(3) trees	(4) meta.	(5) lazy.	(6) lazy.	(7) lazy.	(8) bayes	(9) rules	(10) rule	(11) rule	(12) rule	(13) rule	(14) rule	(15) misc	(16) misc
breast-cancer	(10) 75.54	73.76	68.55	70.28	65.74 *	73.09	72.11	71.7	70.3	65.74 *	72.75	65.1 *	70.95	71.33	69.95	67.14 *
wisconsin-breast-cancer	(10) 94.56	95.85	92.42	94.85	95.28	96.42	96.42	95.99	65.52 *	92.71	95.42	95.99	95.42	93.85	88.56 *	95.7
horse-colic	(10) 85.3	84.5	81.52	81.26	81.27	83.14 *	83.96	78 *	63.05 *	81.52	81.23	80.45	84.22	84.77	61.97 *	78.54 *
credit-rating	(10) 86.09	85.51	85.51	84.64	81.16	85.94	85.07	77.68 *	55.51 *	85.51	85.07	82.61	85.8	85.36	44.64 *	84.64
german_credit	(10) 70.5	72.4	70	69.5	72	74	73.1	75.4 v	70	66.9	72.2	70.5	71.7	70.2	69.9	70.9
pima_diabetes	(10) 73.83	72.93	71.87	74.35	70.17	71.1	73.44	76.31	65.11 *	72.26	73.31	73.97	76.04	75.27	65.49 *	63.94 *
cleveland-14-heart-diseas	(10) 77.85	80.47	71.55	82.11	76.22	82.46	82.13	83.47	54.45 *	71.55	76.88	80.86	81.45	81.84	55.44 *	79.83
hungarian-14-heart-diseas	(10) 81.07	79.95	79.97	77.95	76.83	83.39	81.95	83.7	63.95 *	78.63	78.6	79.64	78.95	80.68	64.28 *	83.01
heart-statlog	(10) 76.67	78.52	72.59	80	75.19	81.48	82.22	83.7	55.56 *	71.11	82.96 v	78.15	78.89	73.33	57.04 *	80
hepatitis	(10) 83.79	76.25	77.5	82.54	80.63	82.63	83.79	84.46	79.38	81.25	81.04	84.42	78	84.46	64.96 *	85.08
ionosphere	(10) 91.46	93.17	82.62 *	90.9	86.33 *	84.89	82.9 *	82.62 *	64.1 *	80.92 *	89.46	90.04	89.75	91.75	35.9 *	94.32
kr-vs-kp	(10) 99.44	95.65 *	66.05 *	93.84 *	89.96 *	95.06 *	97.56 *	87.89 *	52.22 *	66.46 *	97.65 *	98.53 *	99.19	99.06	54.1 *	88.2 *
labor	(10) 73.67	79.33	80	87.33	82.67	91.67	91.67	90	64.67	75.33	77	77.33	77	78.67	86	84.67
sick	(10) 98.81	98.06 *	96.55 *	97.19 *	96.18 *	96.66 *	92.6 *	93.88 *	96.34 *	97.77	96.9 *	98.22	98.62	93.85 *	65.85 *	
sonar	(10) 71.17	77.88	73.05	71.67	86.57 v	75.98	80.83	67.88	53.38 *	62.5	74.5	72.14	73.07	80.31	59.21 *	57.79 *
vote	(10) 96.33	96.32	95.64	95.41	92.42 *	92.89 *	93.36	90.14 *	61.38 *	95.64	95.18	96.09	95.41	94.71	61.38 *	91.74 *
	(v/ /*)	(0/14/2)	(0/13/3)	(0/14/2)	(1/10/5)	(0/12/4)	(0/13/3)	(1/9/6)	(0/4/12)	(0/12/4)	(1/14/1)	(0/13/3)	(0/16/0)	(0/16/0)	(0/3/13)	(0/9/7)

Skipped:

data: UCI binární třídy, klasifikátory: ADTree a univerzální, test: 10-ti násobné křížové ověření, průměrné hodnoty

Analysing: Percent_correct
Datasets: 1
Resultsets: 16
Confidence: 0.05 (two tailed)
Date: 20.5.04 20:25

Dataset	(1) trees.J4	(2) trees	(3) trees	(4) meta.	(5) lazy.	(6) lazy.	(7) lazy.	(8) bayes	(9) rules	(10) rule	(11) rule	(12) rule	(13) rule	(14) rule	(15) misc	(16) misc
1	(160) 83.5	83.78	79.09	83.36	81.79	84.39	84.82	82.6	64.53 *	77.77	83.19	82.67	83.38	84.01	64.54 *	79.46
	(v/ /*)	(0/1/0)	(0/1/0)	(0/1/0)	(0/1/0)	(0/1/0)	(0/1/0)	(0/1/0)	(0/0/1)	(0/1/0)	(0/1/0)	(0/1/0)	(0/1/0)	(0/1/0)	(0/0/1)	(0/1/0)

Skipped:

Key:
(1) trees.J48 '-C 0.25 -M 2' -217733168393644444
(2) trees.ADTree '-B 10 -E -3' -1532264837167690683
(3) trees.DecisionStump ' ' 1618384535950391
(4) meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W trees.DecisionStump' -7378107808933117974
(5) lazy.IB1 ' ' -6152184127304895851
(6) lazy.IBk '-K 10 -W 0' -7902008594385911757
(7) lazy.LWL '-W 0 -K 10 -W bayes.NaiveBayes --' -2788244985131859732
(8) bayes.NaiveBayes ' ' 5995231201785697655
(9) rules.ZeroR ' ' 48055541465867954
(10) rules.OneR '-B 6' -2459427002147861445
(11) rules.DecisionTable '-X 1 -S 5' 2788557078165701326
(12) rules.NNge '-G 5 -I 5' 4084742275553788972
(13) rules.JRip '-F 3 -N 2.0 -O 2 -S 1' -6589312996832147161
(14) rules.PART '-M 2 -C 0.25 -N 3 -Q 1' 8121455039782598361
(15) misc.HyperPipes ' ' -7527596632268975274

(16) misc.VFI '-B 0.6' 8081692166331321866

data: UCI nominální atributy bez chybějících hodnot, klasifikátory: nominální a univerzální, test: 10-ti násobné křížové ověření

Analysing: Percent_correct
Datasets: 7
Resultsets: 16
Confidence: 0.05 (two tailed)
Date: 20.5.04 20:35

Dataset	(1) trees.Id3	(2) trees.	(3) trees.	(4) meta.A	(5) lazy.I	(6) lazy.I	(7) lazy.L	(8) bayes.	(9) rules.	(10) rules	(11) rules	(12) rules	(13) rules	(14) rules	(15) misc.	(16) misc.	
audiology	(10) 79.62		78.32	46.46 *	46.46 *	74.74	58.79 *	77.79	71.23	25.2 *	46.46 *	67.65	75.12	73.87	79.62	65.89 *	52.04 *
breast-cancer	(10) 57.01		75.54 v	68.55 v	70.63 v	65.04 v	73.09 v	72.8 v	72.06 v	70.3 v	65.74	66.48 v	72.75 v	65.79	69.63 v	69.95 v	67.14 v
kr-vs-kp	(10) 99.69		99.44	66.05 *	93.84 *	89.96 *	95.06 *	97.56 *	87.89 *	52.22 *	66.46 *	98.44 *	97.65 *	98.53 *	99.06 *	54.1 *	88.2 *
mushroom	(10) 100		100	88.68 *	96.2 *	100	99.91	100	95.57 *	51.8 *	98.52 *	100	100	100	100	99.77	99.85 *
primary	(10) 33.61		40.11 v	23.3 *	23.3 *	35.64	42.47 v	40.97 v	46.89 v	24.79 *	28.92	36.84	39.22	36.86	38.61	24.79 *	27.16
soybean	(10) 89.88		92.39	19.92 *	19.92 *	91.64	89.01	92.96 v	92.08	13.47 *	33.53 *	85.35 *	87.25	91.94	91.5	86.39	82.41 *
vote	(10) 93.1		96.33 v	95.64	96.09 v	92.43	92.9	93.57	90.14	61.38 *	95.64	94.02	94.73	95.18	95.41 v	61.38 *	91.06
	(v/ /*)		(3/4/0)	(1/1/5)	(2/0/5)	(1/5/1)	(2/3/2)	(3/3/1)	(2/3/2)	(1/0/6)	(0/3/4)	(1/4/2)	(1/5/1)	(0/6/1)	(2/4/1)	(1/2/4)	(1/2/4)

Skipped:
Key:

data: UCI nominální atributy bez chybějících hodnot, klasifikátory: nominální a univerzální, test: 10-ti násobné křížové ověření,
průměrné hodnoty

Analysing: Percent_correct
Datasets: 1
Resultsets: 16
Confidence: 0.05 (two tailed)
Date: 20.5.04 20:36

Dataset	(1) trees.Id		(2) trees	(3) trees	(4) meta.	(5) lazy.	(6) lazy.	(7) lazy.	(8) bayes	(9) rules	(10) rule	(11) rule	(12) rule	(13) rule	(14) rule	(15) misc	(16) misc	
1	(70)	78.99		83.16	58.37 *	63.78	78.49	78.75	82.24	79.41	42.74 *	62.18 *	78.39	80.96	80.31	81.98	66.04	72.55
	(v/ / *)			(0/1/0)	(0/0/1)	(0/1/0)	(0/1/0)	(0/1/0)	(0/1/0)	(0/1/0)	(0/0/1)	(0/0/1)	(0/1/0)	(0/1/0)	(0/1/0)	(0/1/0)	(0/1/0)	(0/1/0)

Skipped:

Key:

(1) trees.Id3 '' -2693678647096322561
(2) trees.J48 '-C 0.25 -M 2' -217733168393644444
(3) trees.DecisionStump '' 1618384535950391
(4) meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W trees.DecisionStump' -7378107808933117974
(5) lazy.IB1 '' -6152184127304895851
(6) lazy.IBk '-K 10 -W 0' -7902008594385911757
(7) lazy.LWL '-W 0 -K 10 -W bayes.NaiveBayes --' -2788244985131859732
(8) bayes.NaiveBayes '' 5995231201785697655
(9) rules.ZeroR '' 48055541465867954
(10) rules.OneR '-B 6' -2459427002147861445
(11) rules.Prism '' 1310258880025902106
(12) rules.DecisionTable '-X 1 -S 5' 2788557078165701326
(13) rules.NNge '-G 5 -I 5' 4084742275553788972
(14) rules.PART '-M 2 -C 0.25 -N 3 -Q 1' 8121455039782598361
(15) misc.HyperPipes '' -7527596632268975274
(16) misc.VFI '-B 0.6' 8081692166331321866

Poznámka: Z testování byly vynechány metody BayesNetK2, BayesNetB a JRip, protože opakovaně havarovaly.

data: Agrodata, klasifikátory: univerzální, test: 10-ti násobné křížové ověření

Analysing: Percent_correct
Datasets: 6
Resultsets: 15
Confidence: 0.05 (two tailed)
Date: 20.5.04 20:26

Dataset	(1) trees.J4	(2) trees	(3) meta.	(4) lazy.	(5) lazy.	(6) lazy.	(7) bayes	(8) rules	(9) rules	(10) rule	(11) rule	(12) rule	(13) rule	(14) misc	(15) misc
eucalyptus	(10) 63.86	49.73 *	49.73 *	51.65 *	54.63 *	59.11	55.58 *	29.08 *	58.16	59.24	53.66 *	61.03	57.87	29.48 *	51.79 *
grub-damage	(10) 33.54	33.46	33.46	39.46	43.96	38.17	45.79 v	31.67	42.08 v	38.88	38.29	36.08	32.58	42.5	45.79 v
pasture-production	(10) 78.33	61.67	71.67	71.67	71.67	69.17	74.17	28.33 *	65.83	75	79.17	73.33	70	71.67	70.83
squash-stored	(10) 65.33	61.67	59.67	73.33	57.67	67.33	61.33	44.33 *	46.67	59.67	39 *	63.67	65.33	61.67	46.67 *
squash-unstored	(10) 82.33	78.33	66.67	65	60.67	68.33	78	42 *	59	74.33	57.33	77	80.33	66.67	56.33 *
white-clover	(10) 64.29	55.48	57.14	63.57	63.81	71.67	60.48	60.48	59.52	60.71	55.95	65.24	63.81	58.81	70
	(v/ /*)	(0/5/1)	(0/5/1)	(0/5/1)	(0/5/1)	(0/6/0)	(1/4/1)	(0/2/4)	(1/5/0)	(0/6/0)	(0/4/2)	(0/6/0)	(0/6/0)	(0/5/1)	(1/2/3)

Skipped:

Key:

data: Agrodata, klasifikátory: univerzální, test: 10-ti násobné křížové ověření, průměrné hodnoty

Analysing: Percent_correct
Datasets: 1
Resultsets: 15
Confidence: 0.05 (two tailed)
Date: 20.5.04 20:26

Dataset	(1) trees.J4	(2) trees	(3) meta.	(4) lazy.	(5) lazy.	(6) lazy.	(7) bayes	(8) rules	(9) rules	(10) rule	(11) rule	(12) rule	(13) rule	(14) misc	(15) misc
1	(60) 64.62	56.72	56.39	60.78	58.73	62.3	62.56	39.31 *	55.21	61.31	53.9	62.73	61.66	55.13	56.9
	(v/ /*)	(0/1/0)	(0/1/0)	(0/1/0)	(0/1/0)	(0/1/0)	(0/1/0)	(0/0/1)	(0/1/0)	(0/1/0)	(0/1/0)	(0/1/0)	(0/1/0)	(0/1/0)	(0/1/0)

Skipped:

Key:

(1) trees.J48 '-C 0.25 -M 2' -217733168393644444
(2) trees.DecisionStump '' 1618384535950391
(3) meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W trees.DecisionStump' -7378107808933117974
(4) lazy.IB1 '' -6152184127304895851
(5) lazy.IBk '-K 10 -W 0' -7902008594385911757
(6) lazy.LWL '-W 0 -K 10 -W bayes.NaiveBayes --' -2788244985131859732
(7) bayes.NaiveBayes '' 5995231201785697655
(8) rules.ZeroR '' 48055541465867954
(9) rules.OneR '-B 6' -2459427002147861445
(10) rules.DecisionTable '-X 1 -S 5' 2788557078165701326
(11) rules.NNge '-G 5 -I 5' 4084742275553788972
(12) rules.JRip '-F 3 -N 2.0 -O 2 -S 1' -6589312996832147161
(13) rules.PART '-M 2 -C 0.25 -N 3 -Q 1' 8121455039782598361
(14) misc.HyperPipes '' -7527596632268975274
(15) misc.VFI '-B 0.6' 8081692166331321866

data: Wekadata, klasifikátory: univerzální, test: 10-ti násobné křížové ověření

```
Analysing: Percent_correct
Datasets: 3
Resultsets: 15
Confidence: 0.05 (two tailed)
Date: 20.5.04 20:29
```

[illegible]

Skipped:

Key :

data: Wekadata, klasifikátory: univerzální, test: 10-ti násobné křížové ověření, průměrné hodnoty

```
Analysing: Percent_correct
Datasets: 1
Resultsets: 15
Confidence: 0.05 (two tailed)
Date: 21.5.04 7:20
```

[illegible]

Skipped:

```
(1) trees.J48 '-C 0.25 -M 2' -217733168393644444
(2) trees.DecisionStump '' 1618384535950391
(3) meta.AdaBoostM1 '-P 100 -S 1 -I 10 -W trees.DecisionStump' -7378107808933117974
(4) lazy.IB1 '' -6152184127304895851
(5) lazy.IBk '-K 10 -W 0' -7902008594385911757
(6) lazy.LWL '-W 0 -K 10 -W bayes.NaiveBayes -' -2788244985131859732
(7) bayes.NaiveBayes '' 5995231201785697655
(8) rules.ZeroR '' 48055541465867954
(9) rules.OneR '-B 6' -2459427002147861445
(10) rules.DecisionTable '-X 1 -S 5' 2788557078165701326
(11) rules.NNge '-G 5 -I 5' 4084742275553788972
(12) rules.JRip '-F 3 -N 2.0 -O 2 -S 1' -6589312996832147161
(13) rules.PART '-M 2 -C 0.25 -N 3 -Q 1' 8121455039782598361
(14) misc.HyperPipes '' -7527596632268975274
(15) misc.VFI '-B 0.6' 8081692166331321866
```