OŠETŘENÍ CHYBĚJÍCÍCH HODNOT V LISP-MINER

Dokumentace

Předmět

4IZ460 – Pokročilé přístupy k DZD

Daniil Bladyko, Michael Drdlíček

xblad11@vse.cz, qdrdm00@vse.cz

CÍL SOFTWARE

Umožnit uživateli získat kompletní soubor dat z tabulky, která původně obsahovala X-categorie (prázdné buňky, ?, -, N/A, not set).

ÚČEL SOFTWARE

Datová sada, kterou uživatel dostane na výstupu, může být použita pro další analýzu (pokud analytik chce aplikovat metody DZD, které nemohou fungovat efektivně při existence chybějících hodnot).

Zároveň výstup může být užitečný také v případě, kdy naléhavě potřebujeme mít kompletní data, ale není vyžadováno, aby všechny hodnoty byly na 100 % pravdivé[1].

POPIS ŘEŠENÉHO PROBLÉMU

Tento software hledá chybějící hodnoty v databázi a na základě požadavků uživatele provede jejich doplnění podle algoritmu vybraného uživatelem. Je to obecnější problém, který se může objevit v jakémkoliv souboru dat.

STRUKTURA PROGRAMU

Software se spouští skriptem *_Main.lua* z příkazové řádky (více v kapitole Instalace).

Adresář, ve kterém se nachází hlavní skript, musí dále obsahovat následující složky:

- Data – musí obsahovat vstupní data, na kterých chce uživatel provést doplnění
- Databases – obsahuje vygenerovanou databázi a metabázi
- Output – obsahuje výstupní data (viz. Výstupy)
- Reports – obsahuje HTML soubor (viz. Výstupy)
- Scripts – obsahuje veškeré skripty použité při běhu programu (viz. Běh programu)

INSTALACE A SPUŠTĚNÍ PROGRAMU

Program se spouští z příkazové řádky. K jeho spuštění je potřeba program *LMExec*. Ten je ke stažení zde: http://lispminer.vse.cz/lmcl/lmexec.html. Vstupní data ve formátu TXT nebo CSV je potřeba uložit do adresáře Data. Pro správný běh programu je potřeba zadat parametry, podle kterých se bude samotné doplňování chovat.

Parametry programu

- *Input* – absolutní cesta k hlavnímu skriptu *(_Main.lua*)
- *databaseName* – název vstupního souboru (z adresáře Data) bez přípony
- *algorithm* – algoritmus, který chceme použít při doplňování dat
    - *simpleStats* – program provede nejjednodušší doplnění podle průměru, mediánu a nebo modusu
    - *ETree* – doplnění podle rozhodovacího stromu
    - *MCluster* – doplnění pomocí shlukové analýzy
    - @all - program spustí všechny algoritmy; více v popisu algoritmů
- *funcType* – musí být vyplněn pouze při výběru algoritmu *simpleStats*

[1] Např. máme tabulku s 50% podílem chybějících hodnot ve sloupci „Pohlaví“. Management od nás vyžaduje, abychom vypočítali relativní počet žen mezi klienty. Půlka hodnot pravděpodobně nebude postačující pro odvození situace v základním souboru hodnot v tomto sloupci, takže je vhodné použit techniky DZD pro vyplnění chybějících hodnot.

    - *avg* – doplnění proběhne na základě průměru v daném sloupci
    - *mdn* – k doplnění bude použit medián sloupce
    - *mod* – použije se módus, defaultní hodnota pokud není uveden
- *columnNames* – jména všech sloupců, které chceme doplnit; jsou 2 možnosti
    - seznam všech sloupců, které chceme doplnit, jednotlivé sloupce oddělujeme čárkou
    - *@allWithMissedValues* – doplnění proběhne u všech sloupců s prázdnými hodnotami, jedná se o defaultní hodnotu, pokud parametr není nastaven
- *output* – volitelný parametr určující obsah výstupního souboru, při jeho nenastavení bude na výstupu primární klíč a všechny sloupečky, ve kterých proběhlo nějaké doplnění. Pokud parametr nastavíme jako *@all,* tak bude navíc ve výstupním souboru i celý vstupní
- *inputFileFormat* – určuje formát vstupních dat, pokud není uveden je jako vstupní formát použit .txt soubor

Parametr *Input* je pro program *LMExec*. Všechny ostatní parametry jsou však pro tento program a proto před každým z nich musí být klíčové slovo "ScriptParam:".

Ukázka nastavení parametrů

Předpokládáme, že se nacházíme v adresáři, ve kterém je umístěn program *LMExec*.

LMExec.exe /Input:C:/cesta/_Main.lua /ScriptParam:databaseName=Hotel /ScriptParam:algorithm=simpleStats /ScriptParam:funcType=mdn /ScriptParam:columnNames=@allWithMissedValues /ScriptParam:output=@all

Na souboru s názvem Hotel bude použito doplnění podle mediánu. Doplněny budou všechny sloupce s prázdnými hodnotami a na výstupu bude vše.

Další ukázkové vstupy jsou k dispozici v souboru "cheat sheet.txt".

VÝSTUPY

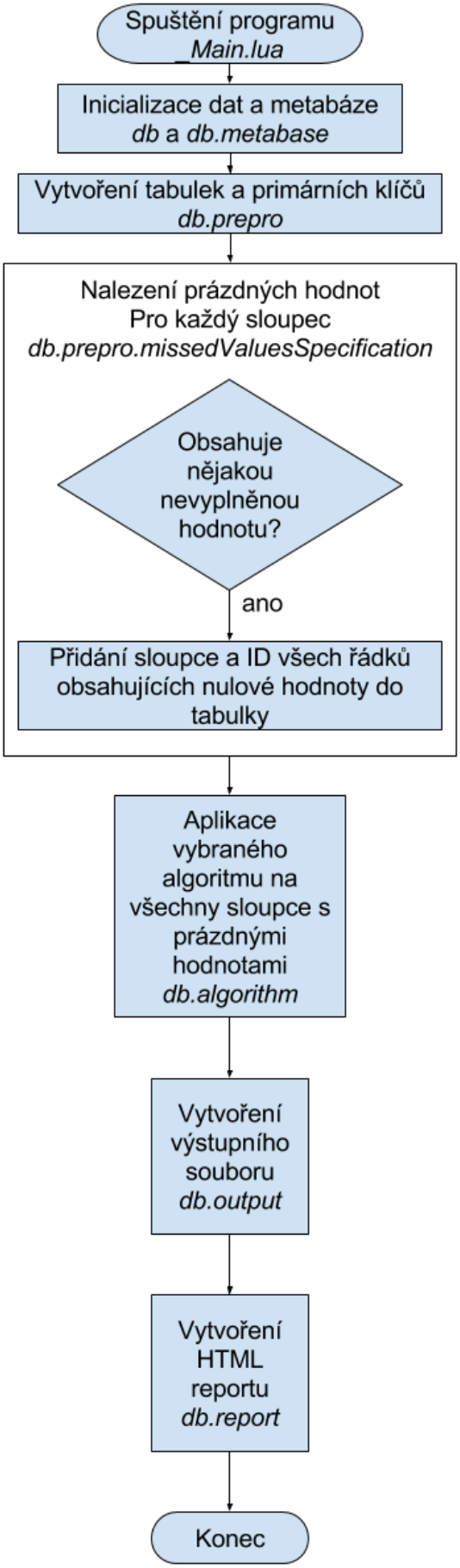
Databáze bez chybějících hodnot (stejný formát jako vstupní data), zpětně použitelný jako soubor externích dat pro aplikace technik DZD. Podle vstupního parametru obsahuje soubor buďto všechny sloupce nebo pouze ty doplněné.

Součástí výstupu je také report ve formátu HTML. Tento soubor obsahuje odkaz na výstupní soubor, počet řádků tabulky, použitou metodu doplnění, ale hlavně informace o tom, kolik doplnění provedl v každém sloupci a primární klíče doplňovaných řádků.

BĚH PROGRAMU

Hlavní běh programu je znázorněn na následujícím vývojovém diagramu. Tento diagram platí pro jakýkoliv z použitých algoritmů.

Nejprve je provedena inicializace a vytváření tabulek. Poté jsou pro všechny sloupec nalezeny prázdné hodnoty. Do tabulky přidáváme sloupce a ID všech řádků s chybějícími hodnotami. Poté je spuštěn uživatelem vybraný algoritmus pro doplnění. Po jeho skončení se vytváří výstupní soubor a HTML report.

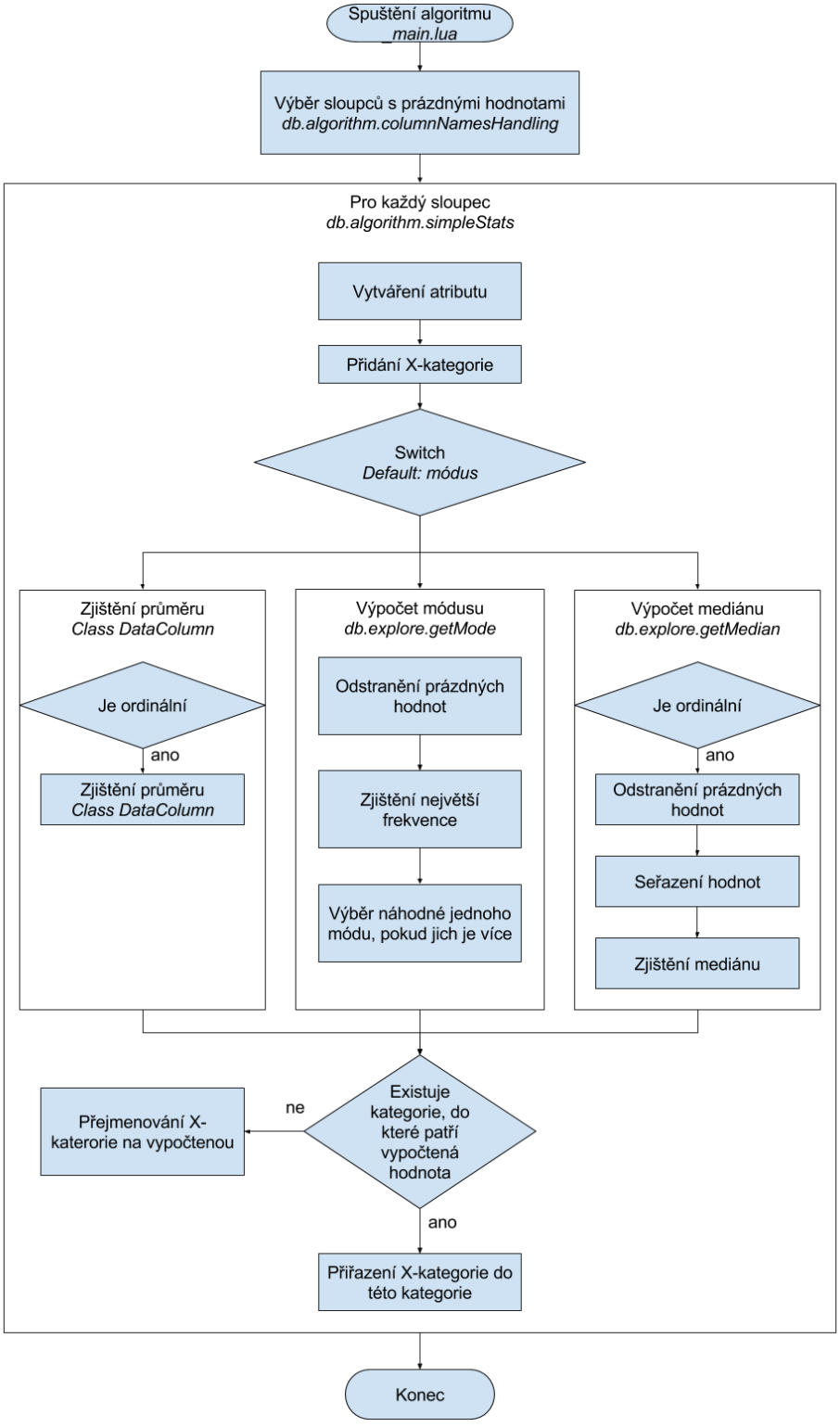

Spuštění programu
_Main.lua
Inicializace dat a metabáze
db a db.metabase
Vytvoření tabulek a primárních klíčů
db.prepro
Nalezení prázdných hodnot
Pro každý sloupec
db.prepro.missedValuesSpecification
Obsahuje nějakou nevyplněnou hodnotu?
ano
Přidání sloupce a ID všech řádků obsahujících nulové hodnoty do tabulky
Aplikace vybraného algoritmu na všechny sloupce s prázdnými hodnotami
db.algorithm
Vytvoření výstupního souboru
db.output
Vytvoření HTML reportu
db.report
Konec

POUŽITÉ ALGORITMY

Jednodušší algoritmy (Metody imputace nezaložené na modelu)

Tyto algoritmy spočívají v dosazení reprezentativní hodnoty do prázdných buněk v daném sloupci, konkrétně se jedná o průměr, medián a modus (pro kategoriální proměnné).

Běh programu je pro všechny tři metody stejný a v následujícím diagramu se liší pouze výpočet dosazované hodnoty.



Po výběru sloupců s prázdnými hodnotami se pro každý sloupec vykoná cyklus. Ten nejprve vytvoří atribut a přidá X-categoii. Pak podle zadané funkce zjišťuje hodnotu, která se bude doplňovat. Defaultně je nastaven módus, při jeho výpočtu se nejprve odstraní prázdné hodnoty, pak je zjištěna největší frekvence a je-li jich víc, je náhodně vybrán jeden módus. Je-li funkce průměr a hodnota ordinální, pak zjistíme průměr ze třídy DataColumn. Pokud je funkce medián a sloupec je ordinální, odstraní prázdné hodnoty, seřadí zbývající a zjistí medián. Nakonec zjišťujeme, jestli již máme kategorii, do které spadá vypočtená hodnota, pokud ano tak do ní přiřadíme X-categorii, pokud ne tak z X-categorie vytvoříme novou.

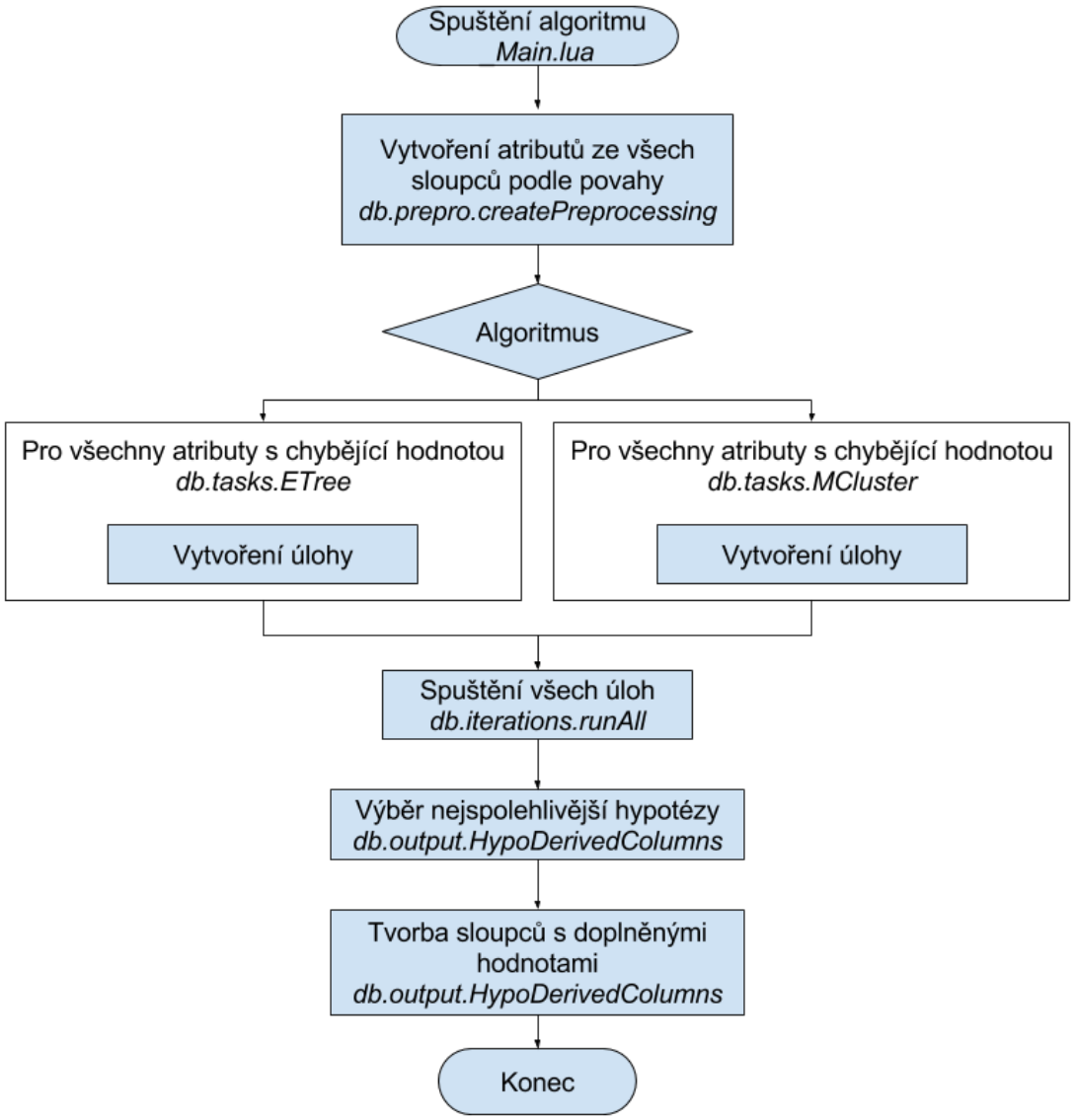
Sofistikovanější algoritmy (Metody imputace založené na modelu)

Shluková analýza

Pomocí standardních nástrojů *LISp-Miner* vytvoříme shluky podle metody k-Mean, pak na základě přiřazení do konkrétního shluku doplňujeme chybějící hodnotu.

Rozhodovací strom

Pomocí standardních nástrojů *LISp-Miner* najdeme hypotézu s nejvyšší kvalitou. Pak aplikujeme tuto hypotézu na pozorování s chybějícími hodnotami a zjistíme, jaká hodnota se hodí nejvíce.



Průběh programu je pro oba sofistikovanější algoritmy stejný, až na vytváření úlohy. Nejprve se vytvoří atributy ze všech sloupců podle povahy, následně se podle zadaného algoritmu pro každý atribut s chybějící hodnotou vytvoří úloha. Poté jsou spuštěny všechny úlohy a po skončení je vybrána nejspolehlivější hypotéza. Nakonec se vytváří sloupce s vyplněnými hodnotami, kde se do jednoho sloupce mohou doplňovat i různé hodnoty podle hypotéz.

Běh všech algoritmů

Uživatel může spustit všechny algoritmy najednou. Aby se tak stalo, musí při spuštění zadat do parametru *algorithm* klíčové slovo *@all*. Program poté spustí všechny dostupné algoritmy. Na výstupu je pak provedeno to doplnění, které je stanoveno jako nejlepší (modus nebo medián z doplněných hodnot různými algoritmy, podle povahy proměnné). Výsledky jednotlivých algoritmů jsou rovněž k dispozici.

PŘIDÁNÍ DALŠÍHO ALGORITMU

Tento program je rozšiřitelný o jakýkoliv uživatelem vytvořený algoritmus na doplňování hodnot. Přidávaný algoritmus musí být napsán v jazyce *lua* a musí být přidán do souboru *Algorithm.lua*, ten je umístěn ve složce Scripts. Pokud by se algoritmus zásadně lišil od již naprogramovaných, je potřeba přidat nové funkce i pro tvorbu výstupu. Algoritmus musí být do skriptu přidán jako funkce v následujícím tvaru:

function db.algorithm.JmenoFunkce(inputParams)

Kde jméno funkce je libovolný název a *inputParams* jsou vstupní parametry.

POPIS KÓDU

Zde je popis všech skriptů s nově vytvořenými funkcemi. Převzaté kódy jsou pouze vyjmenovány v další kapitole.

_Main

Hlavní skript, který podle vstupních parametrů provede inicializaci dat a metabáze, spustí vybraný algoritmus.

Algorithm

Obsahuje všechny použité algoritmy, detailní popis jejich běhu je výše. A funkci pro spuštění všech algoritmů najednou.

Base

V tomto skriptu můžeme přidat další konstanty pro x-categorie.

Explore

Obsahuje převzaté kódy pro inicializaci tabulek a funkce pro zjištění mediánu a módusu.

Output

Funkce simpleStat vytváří výstupní tabulku, nad kterou poté zavolá metodu na vytvoření výstupního souboru. Do tabulky vloží nejprve všechny vstupní sloupce (pokud tak uživatel určil parametrem) a poté přidá všechny sloupce, do kterých bylo něco doplněno.

Funkce makeCSV vytvoří CVS soubor z tabulky a uloží ho do výstupní složky.

Funkce HypoDerivedColumns funguje stejně jako funkce simpleStat, ale všechny modifikované atributy vytvoří tak, že pokud není k dispozici původní hodnota, doplní tu, kterou určil rozhodovací strom nebo shluková analýza.

Report
Funkce print vytvoří HTML soubor popsaný v kapitole Výstupy.

Tasks
Obsahuje převzaté kódy z EverMinerSimple a navíc kódy pro rozhodovací strom a shlukovou analýzu.

Utils
Funkce split rozděluje textový řetězec pomocí separátoru a je použita například na vstupní parametr.

PŘEVZATÉ KÓDY

- EverMinerSimple, konkrétně
    - Metabase
    - Prepro
    - Explore
    - Iterations
    - Tasks

OMEZENÍ

- Algoritmy nemá smysl aplikovat na kompletní datové sady.
- Nalezené vztahy vypovídají spíše o způsobu doplnění než o samotných datech (pokud na výstupních datech budou aplikovány metody DZD).
- Ne všechny datové typy lze doplnit (např. Data/Time).
- Data musí obsahovat alespoň jeden úplně vyplnění sloupec. Shluková analýza neumí pracovat s X-categoriemi. Seznam sloupců, podle kterých by se tvořily shluky, by tedy byl prázdný.

PROVEDENÉ TESTY

Program byl testován na počítači HP ProBook 4330s s dvoujádrovým procesorem o frekvenci 2,5GHz a 4GB RAM. Testy byly provedeny na datech Hotel obsahujících 2000 záznamů, z nichž bylo odstraněno 7 hodnot z různých sloupců. Byly otestovány všechny kombinace vstupních parametrů. Zároveň byl změřen čas pro nejdelší možný běh programu (doplnění všech chybějících hodnot a vše včetně vstupu na výstupu). Zde jsou časy pro jednotlivé algoritmy:

- “simpleStats” – všechny 3 metody trvají cca 7 sekund.
- Rozhodovací strom – 20 sekund.
- Shluková analýza – 17 sekund.
- Všechny algoritmy - 40 sekund.

Složitější algoritmy trvají adekvátně déle, ale stále se jedná o rozumné časy.

POUŽITÉ KNIHOVNY

Knihovny:

1. lm
2. lm.data
3. lm.metabase
4. lm.prepro
5. lm.task
6. lm.task.settings
7. standardní knihovny lua (table, string aj).

ZDROJE

[1] PEJČOCH, David. *Metody řešení problematiky neúplných dat* [online]. [cit. 2015-12-06]. Dostupné z: http://www.dataquality.cz/tutorial/tutorial_04.pdf

[2] *LISp-Miner Control Language Reference* [online]. [cit. 2015-12-06]. Dostupné z: http://lispminer.vse.cz/lmcl/

[3] *The Programming language Lua* [online]. [cit. 2015-12-06]. Dostupné z: http://www.lua.org/