Vysoká škola ekonomická v Prahe, LS 2015

4IZ460 – Pokročilé prístupy k dobývaniu znalostí z databáz

Varianta C-predspracovanie dát

vypracoval:

Peter Trcka, xtrcp00

Zadanie

Zadaním semestrálnej práce bolo vytvoriť skript, ktorý bude v rámci predspracovania dát detekovať extrémne hodnoty. Podrobný popis zadania bol vypracovaný v úvodnej správe . Z uvedených postupov boli naprogramované všetky 3 metódy pre detekciu odľahlých pozorovaní (metóda 4Sigma, Grubsova metóda a detekcia pomocou medzikvartivlového rozpätia) súčasne bola tiež implementovaná detekcia odľahlých pozorovaní pre nominálne premenné podľa princípu relatívnych četností.

V nasledujúcich kapitolách bude podrobne popísaný štruktúra, priebeh, ovládanie a poznámky na vylepšenia.

Štruktúra programu

K vytvoreniu programu pre automatickú detekciu odľahlých a chybných hodnôt bol po konzultácii s vedúcim práce využitý skript programu kolegu Bilíka z 1.3.2015. Pretože štruktúra programu je logicky dobre ucelená neboli v nej vykonané žiadne zmeny.

Program obsahuje adresár data, do ktorého sú ukladané vstupné dáta, adresár DB, ktorý obsahuje výstupnú databázu a jednotlivé metabáze. Posledným adresárom je report, kde je po skončení skriptu uložený informačná správa pre užívateľa s výsledkami automatického spracovania.

Program je rozdelený do jednotlivých súborov:

- RunMe.lua – obsahuje hlavnú kostru programu, v súbore môže užívateľ modifikovať vstupné dáta
- Base.lua – základná definícia prostredia
- Clearing.lua – obsahuje skript, ktorý čistí dáta
- Explore.lua – inicializácia tabuliek a primárneho kľúča
- Metabase.lua – inicializácia metabáz
- Transformation.lua – tvorba atribútov a X kategórií pre jednotlivé stĺpce tabuľky
- JSON.lua – knižnica pre prácu s externými informáciami o databáze vo formáte .json
- Report.lua – obsahuje skript, ktorý vygeneruje výstupnú správe pre užívateľa

Vstup

Hlavným vstupom pre program je databáza vo formáte csv. Aby mohla byť databáza spracovaná musí sa nachádzať v adresári data. Ak sú k dispozícii externé dáta musia byť uložené vo formáte json nahrané do adresára data. Aby program mohol s dátami pracovať musí byť dodržaná konvencia pomenovania vstupných súborov. Názov súboru databázového súboru je ľubovoľný(názovDatabáze**.csv**), ak majú byť požité aj externé dáta názov súboru json musí byť pomenovaný: názovDatabáze**_info.json.**

Výstup

Výstupom je analytická správa vo formáte html v adresári report, vytvorené databáze a metabáze, ktoré sú uložené v adresári **DB**.

Ovládanie programu

Užívateľ môže program jednoducho ovládať pomocou premennej nazovBD, ktorej hodnotu nastavuje v súbore RunMe.lua. Tento súbor je potom potrebné nahrať do LMExec.exe, ktorý spustí program. Proces detekcie odľahlých pozorovaní je plne automatický a funguje pre rôzne databázy. Výhodou je, že obsluha vyžaduje nulové znalosti LMCL.

Inštalácia

Celá štruktúra programu sa musí nakopírovať do adresáru Exec. Súčaťou programu nie automatická vytváranie priečinkov, preto pred spustením musia existovať adresáre **DB**, **data** a **report** už vytvorené inak program skončí chybou.

Priebeh programu

Priebeh programu bol načrtnutý v úvodnej správe.

Detekcia odľahlých pozorovaní – kardinálne premenné

Grubsov test (Grubsova metóda)

Grubsov test funguje na otestovanie minimálnej a maximálnej hodnoty atribútu. Nie je možné metódu reaplikovať. Princíp detekcie prebieha klasickou metódou testovania hypotéz, tj. výpočet testovacej štatistiky, ktorá sa následne porovnáva s kritickým oborom. Kritický obor v tomto prípade predstavuje $G_\alpha = \frac{N-1}{\sqrt{N}} \sqrt{\frac{t^2_{\frac{\alpha}{2N}}(N-2)}{N-2+t^2_{\frac{\alpha}{2N}}(N-2)}}$, kde $t^2_{\frac{\alpha}{2N}}(N-2)$ je kritická hodnota t-rozdelenia s N-2 stupňami voľnosti. Pretože databáze spravidla obsahujú veľké množstvo záznamov N>30 bolo študentovo rozdelenie nahradené normálnym rozdelením. 97,5% kvantil normálneho rozdelenia je rovný 1,960. Zároveň platí, že pre 97,5% kvantil študentovo rozdelenia konverguje k tejto hodnote zhora bola v algoritme použitá hodnota 2 (namiesto 1,960).

Po určení kritického oboru sa vypočíta testovaná štatistika podľa $G_{min} = \frac{\bar{x}-x_{min}}{\hat{\sigma}}$ a $G_{max} = \frac{x_{max}-\bar{x}}{\hat{\sigma}}$. Ak sa aspoň jedna z testovaných štatistík nachádza v kritickom obore je vytvorený atribút s názvom: názovStĺpca**_Grubs** a do x kategórie sú pridané tie hodnoty, pre ktoré testované štatistiky padli do kritického oboru.

Metóda 4 Sigma

Podobne ako Grubsov test, slúži k odhaleniu odľahlých pozorovaní pre minimálnu a maximálnu hodnotu. Postup určenia extrémnych hodnôt touto metódou bol rozdelený do nasledujúcich krokov. Najprv sa určila smerodajná odchýlka pre daný stĺpec. K výpočtu boli použité postačujúce štatistiky úhrn a úhrn štvorcov hodnôt. Úhrn sa podelil počtom nenullových hodnôt a rozptyl bol určený ako rozdiel priemerného úhrnu štvorcov hodnôt a štvorca priemernej hodnoty. Z rozptylu bola vypočítaná smerodajná odchýlka ako odmocnina z rozptylu. Kritická hodnota pre minimum sa rovná priemerná hodnota stĺpca – 4* smerodajná odchýlka stĺpca. Kritická hodnota pre maximum s rovná priemerná hodnota stĺpca + 4* smerodajná odchýlka stĺpca. Za extrémne hodnoty podľa metódy 4 sigma sú považované tie, ktoré menšie ako minimálna kritická hodnota, alebo väčšie ako maximálna kritická hodnota.

V prípade, že minimálna alebo maximálna hodnota stĺpca boli označené ako extrémne, vytvoril sa atribút s názvom: názovStĺpca_4sigma. Do x kategórie boli pridané tie krajné hodnoty, ktoré ležali mimo kritický interval.

BloxPlot

Oproti predchádzajúcim metódam môže byť metóda BoxPlot použitá pre určenie intervalu extrémnych hodnôt. Postup je nasledovný: Najprv sa spočíta počet zdaných hodnôt. Potom sa hodnoty zoradia vzostupne. V zoradenom súbore a vyberie hodnota pre dolný a horný kvartil. Vypočíta sa medzikvartilové rozpätie ako rozdiel horného a dolného kvartilu. Potom sa určí dolná a horná hodnota pre kritického intervalu. Dolná hodnota je dolný kvartil – 1,5* medzikvartilové rozpätie, horná hodnota je horný kvartil + 1,5* medzi kvartilové rozpätie.

Keď je určený kritický interval porovná sa s ním minimálna a maximálna hodnota stĺpca. Ak aspoň jedna leží mimo kritický interval vytvorí sa atribút s názvom: názovStĺpca_BoxPlot. Do x kategórie sa pridá interval <minimálna hodnota, dolná hranica interval> príp <horná hranica intervalu, maximálna hodnota> v závislosti na tom, či sa nachádzali mimo kritický interval minimálna maximálna alebo obe naraz.

Detekcia odľahlých pozorovaní – nominálne premenné

Metóda relatívnych četností

Metóda relatívnych četností nemôže byť označená ako všeobecne uznávaný prístup k detekcii odľahlých pozorovaní nominálnych premenných. Princíp detekcie funguje na základe relatívnych četností. Najprv sa určí počet nenullových hodnôt pre daný atribút. Potom sa určí minimálna hranica četností ako napr. 0,5% z nenullových hodnôt (n). Každá reťazec, ktorého četnosť je menšia ako 0,005*n je označený ako chybný a je presunutý v rámci daného atribútu do x kategórie.

Záver

Jazyk LMCL je veľmi dobre zdokumentovaný. Chvíľku mi trvalo kým som si naň zvykol, ale vďaka praktickým príkladom to nebola až tak náročná úloha. Výsledný skript sa zhoduje so zadaním v úvodnej správe.

Jednotlivé metódy väčšine prípadov označovali rovnaké hodnoty ako extrémy. V prípade, že žiadna metóda pre daný stĺpec neidentifikovala extrémne hodnoty užívateľ sa atribútom za hľadiska odľahlých hodnôt zaoberať nemusí. Ak ale určité hodnoty ako extrémne identifikované boli, odporúčam užívateľovi X kategóriu skontrolovať.

Pri výpočtoch som ocenil metódy triedy DataColumn konkrétne getAvg(), getMin(), getMax(). Pretože sa pracuje s objektom v ktorom sú uložené dáta chýbali mi tu ostatné metódy, ktoré by vracali ďalšie základné štatistiky. Pre inšpiráciu navrhujem rozšíriť triedu DataColumn o nasledujúce metódy:

getVar() – metóda vracia rozptyl dát

getQuantile(integer quantil) – metóda vracia hodnotu kvantilu

getSkewness() – metóda vracia šikmosť dát

getKrutosis() – metóda vracia špicatosť

getSum(integer level) – metóda vracia úhrn hodnôt ^ level

getNoNullCount() – metóda vráti počet nenullových hodnôt v stĺpci

getCount() – metóda vráti počet hodnôt v stĺpci

Príslušné hodnoty je možné na základe už existujúcich metód samozrejme dopočítať ale implementácia uvedených metód by zefektívnila a sprehľadnila tvorbu LMCL skriptu.