Data z ústředen - modelování úspěšnosti hovoru

Michal Trs

***Abstrakt*— Pokus o zpracování dat z telefonní ústředny pomocí neuronové sítě**

I. Zadání

Sledování poměru počtu úspěšných a neúspěšných hovorů v daných časových intervalech pro jednotlivé dni.

II. Úvod

A. Předzpracování dat

Před „nasypání“ dat do neuronové sítě je nutno je předzpracovat. Pro předzpracování dat jsem napsal několik scriptů. Zde uvádím pouze jejich názvy a krátký popis. Popis ovládání naleznete v kapitole VI.

- **rawtoweeks.sh** rozdělí data z ústředny do souborů po týdnech
- **preprocess.sh** vybere z původních dat potřebné sloupce, převede datum, přečísluje směry příchozích / odchozích hovorů a upraví výstup na úspěšný (1) / neúspěšný (0) hovor
- **convert.sh** nahradí sloupec 5 - úspěšný / neúspěšný hovor procentuální úspěšností hovorů v danou hodinu
- **reduct.sh** redukuje počet dat
- **toNNS.sh** převede data pro simulátor JavaNNS
- **toWeka.sh** převede data do formátu Weka

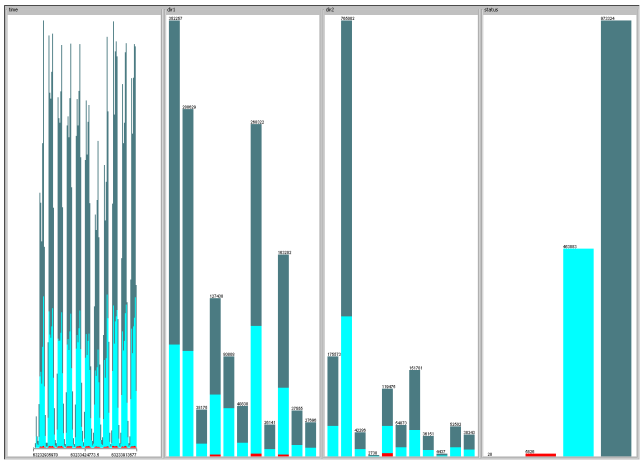
Z dat z ústředny nás pro sledování úspěšnosti hovorů zajímá původní sloupec `E` - datum a sloupec `M` - způsob ukončení hovoru. Datum z lineárního tvaru jsem rozdělil do dvou sloupců - den v týdnu a hodinu ve dni. Takto bychom měli pouze dva vstupy do sítě a to je málo, proto ještě přidáme čísla ústředen příchozích a odchozích hovorů (původní sloupec `K` a `L`). Z těchto čtyř vstupů a jednoho výstupu se pokusíme naučit neuronovou síť predikovat podle času a směru hovorů jejich úspěšnost.

B. Kontrola - histogram

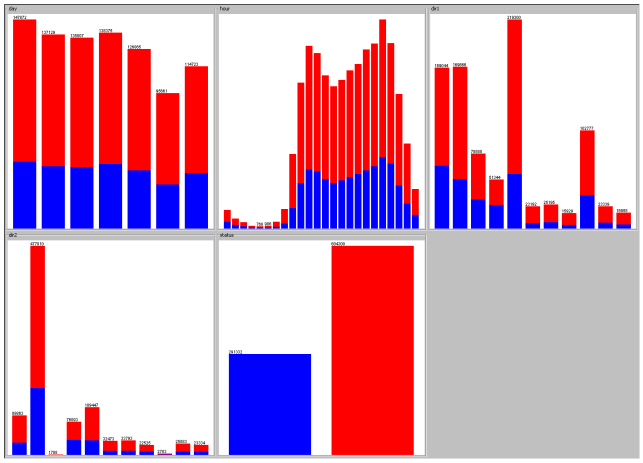
Pro kontrolu zda scripty pracují správně a zachovávají stejné rozložení dat si převedeme původní data (nebo po průchodu scriptem `rawtoweeks.sh`) programem Sumatra TT2 do formátu Weka. Z výstupu si zobrazíme např programem Weka nebo Sumatra TT2 histogram a porovnáme jej s výstupem ze scriptu `reduct.sh`.

III. Cíl práce

Cílem této práce by mělo být sledování a predikování úspěšnosti hovoru z dat získaných z telefonních ústředen. Zadání není příliš konkrétní, neříká z jakého sloupce lze rozpoznat, zda byl hovor úspěšný.



Obr. 1. Originální data



Obr. 2. Kompletní data - výstup scriptu prepocess.sh

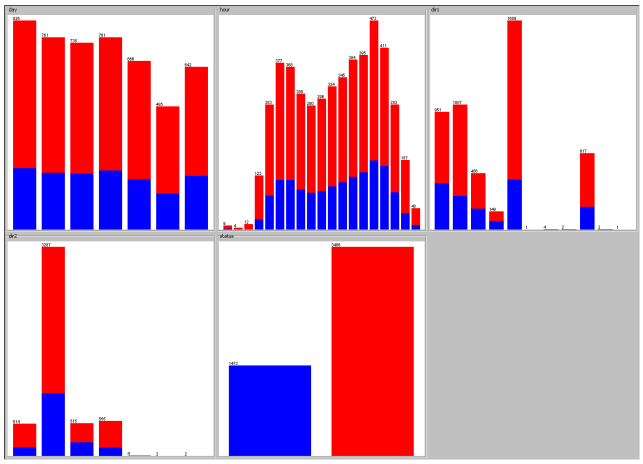
IV. Experimenty

A. Problémy v zadání

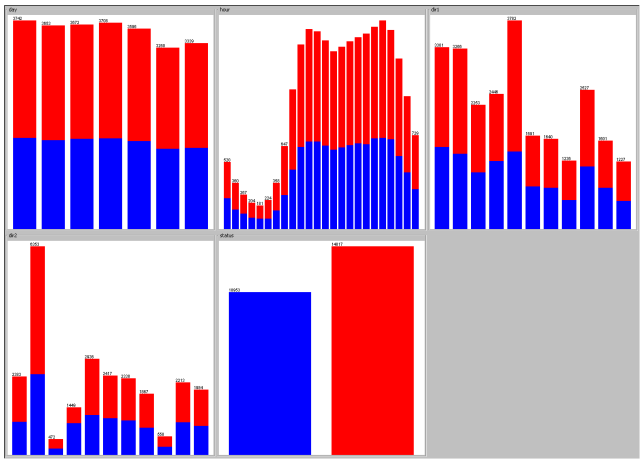
V souboru `format_description.txt` je napsáno, že sloupec E je čas vzniku hovoru v sekundách od 1.1.1970. Standardní unixovou funkcí jsem tedy převedl číslo na datum a vyšel rok kolem 4000. Proto datum převádím ručně na den v týdnu a hodinu ve dni.

B. Předzpracování

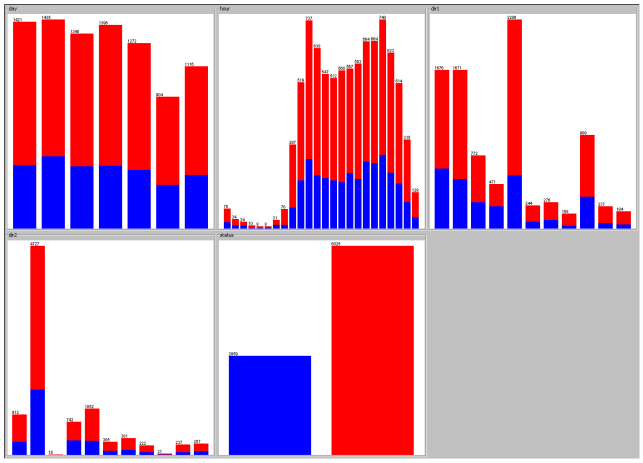
První, ne málo náročný úkol bylo předzpracování dat. Vybral jsem soubory `041*.txt` z adresáře `data`. Na obrázku 1 jsou vidět čtyři sloupce (zleva: čas v lineárním tvaru, příchozí směr, odchozí směr, způsob ukončení hovoru)z původních dat, které budeme dále zpracovávat. Všimněme si času v lineárním



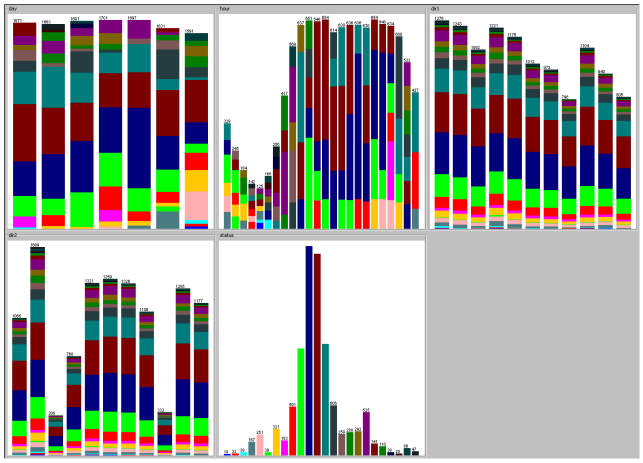
Obr. 3. Redukovaná data - potlačení málo zastoupených dat



Obr. 4. Redukovaná data - posílení málo zastoupených dat



Obr. 5. Náhodně vybraná data



Obr. 6. Redukovaná data - na výstupu procentní úspěšnost hovoru

tvaru. Je patrné, že má 11 lokálních maxim, což představuje 11 dní.

Abychom z dat dostali nějaký konkrétní časový interval, použijeme následující script `./rawtoweeks.sh data/0410*.txt`. Tím nám vzniknou 3 soubory. Podle velikosti, nebo náhledem na rozložení dat vybereme soubor, ve kterém je obsažen celý týden. V našem případě je to soubor `104552.raw`.

Na tento soubor pustíme script `./preporcess.sh 104552.raw > telecom.data`. Histogram takto upravených dat je na obrázku 2. Význam sloupců je následující (zleva): den v týdnu, hodina ve dni, příchozí směr, odchozí směr, způsob ukončení hovoru. Nenechte se zmást jiným rozložením dat u příchozího a odchozího směru. Ve skutečnosti se nezměnilo, je to dáno přečíslováním ústředen, které se dá dohledat v souboru `telecom_report.txt`. Tyto data pro učení neuronové sítě ještě nelze použít, protože obsahují necelých 900 000 učicích vzorů a to je moc.

C. Redukovaná data

První možnost jak zmenšit počet dat na únosnou míru je jejich redukce podle histogramu. To za nás udělá script `./reduct.sh [-i] [-r] telecom.data > soubor.txt`. V tabulce I jsou jména souborů, která jsem použil. Význam parametrů je popsán v kapitole VI.

parametr	název souboru	číslo obrázku
–	telecom_dec.txt	3
-i	telecom_inc.txt	4
-r	telecom_rnd.txt	5

TABULKA I
POUŽITÉ NÁZVY SOUBORŮ

Dále je nutné soubory převést do formátu pro zvolený simulátor neuronové sítě. K tomu opět využije script, tentokrát `toWeka.sh -o telecom_dec.arff telecom_dec.txt`. Takto převedeme i zbylé dva soubory.

Tyto soubory postupně otevřeme v programu Weka a pro každý z nich použijeme klasifikaci pomocí MultilayerPerceptron (MLP), RBFNetwork (RBF) a SMO. Použil jsem defaultní nastavení, jen v případě RBFNetwork jsem nastavil parametr numClusters na 20. Pro validaci byla použita 10x cross-validace. Procentuální počet správně klasifikovaných vzorů je shrnut v tabulce II.

%	MLP	RBF	SMO
telecom_dec.arff	65.2	69.4	69.8
telecom_inc.arff	53.7	56.5	55.3
telecom_rnd.arff	67.2	67.0	67.4

TABULKA II
VÝSLEDKY KLASIFIKACE - BINÁRNÍ VÝSTUP

D. Procentuální úspěšnost hovoru

Protože výsledky z redukovaných dat nejsou valné, nahradil jsem v předzpracování krok popisovaný v kapitole IV-C. V každém dnu pro jednotlivé hodiny vypočteme procentní úspěšnost hovorů a nahradíme s ní informaci zda byl hovor úspěšný (5.sloupec). Takto redukujeme data a zároveň dáme více informací na výstup. Opět na tento krok použijeme script a rovnou výstup převedeme do formátu Weka. Použití je následující:

```
$./convert.sh telecom.data | \
./toWeka -o telecom_per.arff
```

Tento soubor načteme v programu Weka a použijeme klasifikace jako v kapitole IV-C. Testování bylo časově náročné a v případě sítě RBF končilo pádem aplikace. Procentuální počet správně klasifikovaných vzorů je shrnut v tabulce III.

%	MLP	RBF	SMO
telecom_per.arff	91.8	N/A	82.5

TABULKA III
VÝSLEDKY KLASIFIKACE - PROCENTUÁLNÍ VÝSTUP

V. DISKUSE

Nejprve jsem experimentoval v simulátoru JavaNNS. Pomocí scriptů, popsaných v kapitole VI, jsem předzpracoval data. Téměř libovolná backpropagation síť nebyla schopna se daný problém naučit. To je dáno tím, že žádný vstupní parametr nám neklasifikuje binární výstup, jak je vidět na obrázku 2.

Dále jsem takto předzpracovaná data zkoušel klasifikovat v programu Weka. Použil jsem sítě MLP, RBF a SMO. Výsledky pro různý výběr z původních dat jsou shrnuty v tabulce II. Ukázalo se, že na výběru (redukci) dat a použité neuronové síti téměř nezáleží. Je to dáno tím, že v původních datech ani jeden parametr neklasifikuje daný problém a proto tento postup nevede k uspokojivým výsledkům.

Proto jsem přistoupil k dalšímu zpracování (viz kapitola IV-D). Tímto jsem binární výstup nahradil nespojitou informací jak byly v danou hodinu hovory úspěšné. Nyní učíme síť tento problém: Na vstupy je čas a směr hovoru a chceme znát na kolik bude daný hovor úspěšný. Z obrázku 6 je patrné, že tento problém již vstupní parametry částečně klasifikují. Výsledky jsou shrnuty v tabulce III. Dosahujeme řádově lepších výsledků a proto by se další pokusy o zpracování měly ubírat tímto směrem. Bylo by dobré ještě zkusit spojitý výstup, případně přidat na vstup další atributy, nebo určovat úspěšnost hovoru například z jeho délky.

VI. POPIS A OVLÁDÁNÍ VYTVOŘENÝCH SCRIPTŮ

A. rawtoweeks.sh

- **popis:** rozdělí soubory na vstupu podle času po týdnech
- **vstup:** soubor(y) s daty z ústředny
- **výstup:** soubory ve stejném formátu jako vstup, pojmenovány *číslo_týdne.raw*

B. preprocess.sh

- **popis:** Z původních dat vybere sloupce E, K, L a M, přičemž sloupec E rozdělí na 2 sloupce - den v týdnu (0-6) a hodinu (0-23) ve dni. Sloupce K a L přečísluje, aby začínali nulou a v původním sloupci M nahradí hodnotu 3 jedničkou a ostatní hodnoty nulou.
- **vstup:** soubor(y) s daty z ústředny, případně soubor(y) generovaný scriptem *rawtoweeks.sh*
- **výstup:** data na stdout, report o přečíslování směrů hovorů do souboru *telecom_report.txt*

C. reduct.sh

- **popis:** redukuje počet dat a má tyto parametry
  - -i posílí málo zastoupená data
  - -r náhodně vybere data

  bez parametru script potlačí málo zastoupená data
- **vstup:** stdin, nebo soubor ve formátu generovaným scriptem *preprocess.sh*
- **výstup:** stdout - stejný formát jako vstup

D. convert.sh

- **popis:** vypočítá pro každou hodinu v týdnu procentuální úspěšnost hovoru a touto hodnotou nahradí pátý sloupec
- **vstup:** stdin, nebo soubor ve formátu generovaným scriptem *preprocess.sh*
- **výstup:** stdout - stejný formát jako vstup

E. toWeka.sh / toNNS.sh

- **popis:** doplní hlavičku do souboru pro daný formát
- **vstup:** stdin, nebo soubor ve formátu generovaným scriptem *preprocess.sh*
- **výstup:** telecom.arff / telecom.pat, nebo specifikujeme výstupní soubor pomocí parametru *-o jmeno_souboru*

VII. Závěr

Z počátku snadně vypadající úkol jak se jevila predikace úspěšnosti hovoru podle času nebyla vůbec snadná. Většinu času jsem věnoval předzpracování dat a proto je mu v reportu věnováno tolik prostoru. Jak je z výsledků patrné, tak na dobrém výběru atributů a hlavně statistickém předzpracování velice záleží. Pokud použijeme pouze binární výstup, který nám vstupní atributy neklasifikují, tak dosahuje 65% úspěšnosti predikce. Pokud však data dokážeme přepočítat a nalézt tak atributy, kterými je výstup klasifikován, tak v našem případě dosahujeme úspěšnosti 90%.

Poděkování

Děkuji tvůrcům programů Sumatra TT2 a Weka. Dále tvůrcům GNU softwaru za vznik programů awk, cut, cat a další. Bez nich by nebylo možně předzpracování dat.

Dále děkuji za poskytnutí dat firmě Strom Telecom.

Literatura

[1] Šnorek M., *Neuronové sítě a neuropočítače*, skriptum ČVUT, 2002.