Semestrální práce z předmětu m6f

χ^2 test dobré shody

Ikar Pohorský
21. 5. 2006

Zadání

Ověřte, nebo zamítněte hypotézu, že četnost souborů v jednotlivých třídách velikostí odpovídá exponenciálnímu rozložení.

Získání dat

Pro získání dat z libovolného adresáře postačí následující příkaz:

```
ls -lR jmeno_adresare | gawk '/^-/{print $5}' > file_size_list
```

Ten uloží do souboru `file_size_list` velikosti souborů, které najde po cestě adresářem, který dostane jako parametr.

Zpracování dat

Pro rozdělení dat do jednotlivých tříd a výpočet testové statistiky jsem napsal skript v jazyce awk (viz příloha). Spuštění provedeme příkazem:

```
gawk -f compute.awk file_size_list
```

Popis práce skriptu

Ve skriptu je implicitně definován počet tříd na deset tříd a horní mez poslední třídy na 10000, čili 10kB. Tyto parametry mohou dosti ovlivnit výsledek výpočtu, proto je potřeba si s nimi pohrát. Program si navíc přidá jednu třídu (tedy jedenáctou) s rozsahem od horní meze poslední třídy do nekonečna.

Výkonná část skriptu čte vstupní soubor a ukládá do pole počet souborů, jejichž velikost vyhovuje dané třídě. Každá položka v poli reprezentuje jendu třídu.

Na závěr skript provede výpočet statistiky a na standatrní výstup vypíše tabulku.

Ukázkový výstup:

```
$ ls -lR /etc | gawk '/^-/{print $5}' | gawk -f compute.awk
-------------------------------------------------------------------------------
| <      od ; do      ) |    o_i |    p_i |        e_i | (o_i-e_i)^2 / e_i |
-------------------------------------------------------------------------------
| <       0 ; 1100    ) |    350 | 0,2516 |    160,502 |           223,733 |
| <    1100 ; 2200    ) |    119 | 0,1883 |    120,124 |             0,011 |
| <    2200 ; 3300    ) |     56 | 0,1409 |     89,905 |            12,786 |
| <    3300 ; 4400    ) |     26 | 0,1055 |     67,287 |            25,334 |
| <    4400 ; 5500    ) |     12 | 0,0789 |     50,360 |            29,219 |
| <    5500 ; 6600    ) |     18 | 0,0591 |     37,691 |            10,287 |
| <    6600 ; 7700    ) |      6 | 0,0442 |     28,209 |            17,485 |
| <    7700 ; 8800    ) |      3 | 0,0331 |     21,112 |            15,539 |
| <    8800 ; 9900    ) |      5 | 0,0248 |     15,801 |             7,383 |
| <    9900 ; 11000   ) |      4 | 0,0185 |     11,826 |             5,179 |
| <   11000 ; oo      ) |     39 | 0,0551 |     35,183 |             0,414 |
-------------------------------------------------------------------------------
sum {(o_i - e_i)^2 / e_i} =  347,37
```

Popis sloupců a postup výpočtu

Výstup programu má nezasvěceným nic neříkající záhlaví tabulky o pěti sloupcích a výsledek statistiky.

První sloupec ukazuje meze třídy. Zvláštností je snad pouze poslední třída, která zahrnuje nekonečno. Toto nekonečno je ve výpočtech reprezentováno velkým číslem. Takovým velkým číslem rozumějme číslo, které když bude zvětšeno (např. o řád) již nezmění hodnoty ostatních sloupců.

Druhý sloupec říká, kolik souborů se v dané třídě pro konkrétní adresář vyskytuje. Označme toto číslo jako o_i.

Třetí sloupec je již zajímavější. Pro každou třídu je zde vypočítána teoretická pravděpodobnost výskytu jednoho souboru. Tato hodnota se pro exponenciální rozdělení vypočítá následujícím vzorcem:

$$p_i = \int_a^b \lambda\, e^{-\lambda x}\, dx = [-e^{-\lambda x}]_a^b = e^{-\lambda a} - e^{-\lambda b},$$

kde a je dolní mez třídy, b je horní mez třídy a parametr λ je neznámý. Odhad tohoto parametru provedeme například metodou maximální věrohodnosti a dojdeme ke vztahu:

$$\hat{\lambda} = \frac{1}{\bar{x}} = \frac{n}{m},$$

kde n je počet všech souborů a m je celková velikost všech souborů.

Ve čtvrtém sloupci vidíme, kolik by třída v případě exponenciálního rozdělení teoreticky obsahovala souborů, což vypočítáme vynásobením pravděpodobnosti výskytu jednoho souboru celkovým počtem souborů:

$$e_i = n\, p_i.$$

V posledním sloupci je uvedena odchylka vypočítané a naměřené hodnoty vypočítaná z následujícího vztahu:

$$\frac{(o_i - e_i)^2}{e_i}.$$

Hodnota testové statistiky je uvedena pod tabulkou. Pro úplnost uveďme vzorec:

$$\sum \frac{(o_i - e_i)^2}{e_i}.$$

χ^2 test dobré shody

Abychom mohli zamítnout hypotézu, že četnost souborů v jednotlivých třídách podléhá exponenciálnímu rozdělení, musí platit

$$\sum_{i=i}^{k} \frac{(o_i - e_i)^2}{e_i} > \chi^2_{1-\alpha}(k-1-q)\,,$$

kde α je hladina výzamnosti, na které zamítneme hypotézu a $\chi^2_\beta(\nu)$ je β-kvantil rozdělení o $(k-1-q)$ stupních volnosti, kde k je počet tříd a q je počet odhadovaných parametrů. V našem případě je $q=1$ a na hladině významnosti 5% z tabulek vyhledáme, že

$$\chi^2_{0,95}(k-1-q)=\chi^2_{0,95}(11-1-1)=\chi^2_{0,95}(9)=16{,}919\,.$$

Pokud je hodnota testové statistiky, než $(1-\alpha)$ -kvantil rozdělení χ^2, zamítneme hypotézu, že se data řídí exponenciálním rozdělením.

Závěr

Nepodařlo se mi v systému najít adresář, který by testem prošel bez zamítnutí hypotézy o exponenciálním rozložení dat. Testovány byly adresáře /etc; /usr; /var. I když na první pohled data vypadají, že by exponenciálnímu rozložení mohla odpovídat, test je dokonce i na hladině významnosti 0,5% stále nekompromisní.

Příloha

zdojový kód awk skriptu:

```
BEGIN {
	# kolik chceme mit trid
	resolution=10;

	# horni mez posledni tridy
	last_class_top=6000;

	new_array[resolution + 1]=0;
	files_count=0;
	total_file_size=0;
}
{
	files_count++;
	total_file_size += $1;

	# hledame, do ktere tridy pasuje velikost souboru
	for (i=0; i<resolution; i++) {
		if ($1 >= last_class_top*i/resolution &&
		    $1 < last_class_top*(i+1)/resolution) {
		   new_array[i]++;
		   next;
		}
	}

	# pokud nikam nepasuje, je za nejvyssim odhadem -> posledni trida
	new_array[i]++;
}
END {
	lambda = files_count / total_file_size;

	print "-------------------------------------------------------------------------------"
	printf "| <      od ; do      ) |    o_i |    p_i ";
	printf "|        e_i | (o_i-e_i)^2 / e_i |\n";
	print "-------------------------------------------------------------------------------"

	for (i = 0; i <= resolution; i++) {
		# leva mez tridy
		from = last_class_top*i/resolution;
		# prava mez tridy; pokud je posledni, dosazujeme "nekonecno"
		to = (i==resolution) ? 100000 : last_class_top*(i+1)/resolution;
		printf "| <%8s ; %-8s)", from, (i == resolution) ? "oo" : to;
		# pocet souboru v dane tride
		files = new_array[i] ? new_array[i] : 0;
		printf " | %6s", files;
		# teoreticka pravdepodobnost vyskytu jednoho souboru ve tride
		p_i = exp(-lambda*from) - exp(-lambda*to);
		printf " | %1.4f", p_i;
		# teoreticky pocet souboru v dane tride
		e_i = p_i * files_count;
		printf " | %10.3f", e_i;
		# odchylka namerene vs. vypocitane (o_i - e_i)^2 / (e_i)
		test_stat = (files - e_i)*(files - e_i)/e_i;
		printf " | %17.3f |\n", test_stat;
		# soucet odchylek
		sum += test_stat;
	}
	print "-------------------------------------------------------------------------------"
	# soucet odchylek
	print "sum {(o_i - e_i)^2 / e_i} = ", sum;
}
```