Binomické a Poissonovo rozdelenie

Cieľom tohto textu je ilustrovať priebeh binomického a Poissonovho rozdelenia, naučiť sa pracovať s cyklom v matlabe a v rámci možností racionalizovať výpočtový proces.

<u>Príklad 1:</u>

Do školy v Hornej Matlabovej chodí 49 žiakov. Pravdepodobnosť absencie jedného študenta v ktorýkoľvek deň je 11%. Vypočítajme pravdepodobnosť, že v určitý deň bude chýbať *k* žiakov, pre *k* = 0, 1, 2, … , 49.

Pravdepodobnosť súčasnej absencie *k* žiakov je daná binomickým rozdelením:

$$p(k) = n! / (k! (n\text{-}k)!) * p^k * (1\text{-}p)^{(n\text{-}k)}$$

Tieto hodnoty v matlabe získame najjednoduchšie pomocou cyklu:

```
n = 49; p = 0.11;
for k = 0:n, bp(k+1) = factorial(n)/factorial(k)/factorial(n-k) * p^k * (1-p)^(n-k); end
sum(bp)
bar(0:n,bp)
```

<u>Poznámky:</u>

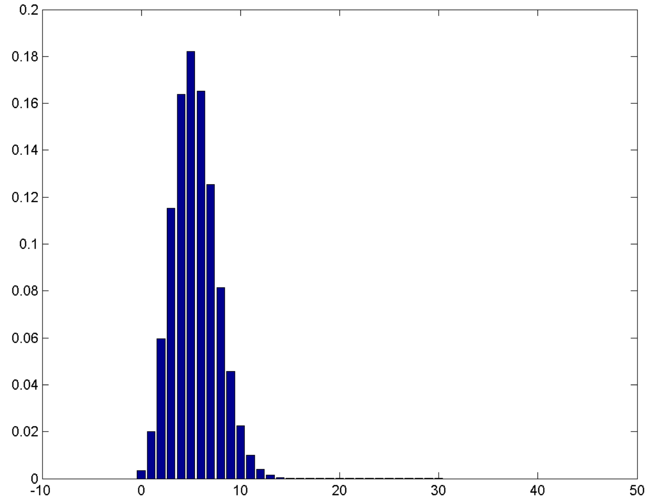
a) Spôsob výpočtu hodnôt bp nie je najvhodnejší a v ďalšom príklade ukážeme, prečo ho treba a ako ho možno vylepšiť.

b) Vektor pravdepodobností bp má dĺžku 50. Keďže Matlab indexuje zložky vektorov zásadne od jednotky vyššie, musíme písať bp(k+1) = … . Hodnotám p(0), p(1), … , p (49) teda zodpovedá bp(1), bp(2), … , bp(50).

c) Za predpisom pre výpočet bp(k+1) musí NUTNE nasledovať bodkočiarka. Ak ju tam nedáme, v každom kroku (je ich 50) Matlab vypíše všetky hodnoty vektora bp, ktoré už vypočítal.

d) Správnosť výsledku skontrolujeme sčítaním hodnôt vektora bp. Výsledok by mal byť 1.

e) Napokon rozdelenie vykreslíme:



Príklad 2:

Na novú Technickú univerzitu v Matlabckách chodí 765 študentov. Kvôli problémom s dochádzaním je pravdepodobnosť absencie študenta v ktorýkoľvek deň 23% . Vypočítajme pravdepodobnosť, že v určitý deň bude chýbať *k* žiakov, pre *k* = 0, 1, 2, … , 765.

Opäť ide o výpočet hodnôt binomického rozdelenia pre n=765 a p=0.23. Ak by sme chceli postupovať podľa predošlého návodu, narazíme na problém – Matlab si neporadí s faktoriálom čísla 765. Najvyšší faktoriál, ktorý zvládne, je 170.

Vďaka tomu, že je Matlab rýchly, sme narazili iba na jeden problém, v skutočnosti sú však dva. Aj ten druhý však treba riešiť.

a) Prvý problém je v hodnotách faktoriálov, ktoré sú príliš veľké. Treba si však uvedomiť, že my nepotrebujeme samostatné hodnoty všetkých troch faktoriálov, ale len výsledný podiel, ktorý nebude tak veľký. Výpočet treba prispôsobiť tak, aby sme sa týmto veľkým číslam vyhli.

b) Druhý problém spočíva v množstve zbytočných (zbytočne opakovaných) operácií. Napr. ak máme vypočítaný (k-1)!, nie je nutné počítať k! od začiatku, ale stačí vziať predošlý výsledok a násobiť ho číslom k. Počet operácií tak dosť podstatne klesne. Hoci je Matlab rýchly a pri jednoduchších výpočtoch nám nedá pocítiť neefektívnosť algoritmu, pri dlhších výpočtoch bude rozdiel citeľný. Avšak bez ohľadu na to, čo počítame, efektívnosť algoritmu je otázkou estetiky a programátorskej cti.

Oba problémy môžeme naraz vyriešiť tak, že každý ďalší člen vektora hodnôt pravdepodobnostnej funkcie budeme počítať z predchádzajúceho člena. Ak platí:

$$p(k) = n! / (k!\,(n-k)!) * p^{k} * (1-p)^{(n-k)}$$
$$p(k-1) = n! / ((k-1)!\,(n-k+1)!) * p^{k-1} * (1-p)^{(n-k+1)}$$

potom

$$p(k) = p(k-1)*(n-k+1)/k*p/(1-p)$$

Cyklus pre výpočet bp bude vyzerať takto:

```
clear bp
n = 765; p = 0.23;
bp(1) = (1-p)^n;
for k=1:n, bp(k+1) = bp(k)*(n-k+1)/k*p/(1-p); end
sum(bp)
bar(0:n,bp), colormap([0.5 0 0])
```

Počet vykonaných operácií je teraz rádovo násobkom n, kým v predošlom prístupe si vyžadoval rádovo n^n operácií, čo je pre väčšie n likvidačná hodnota.

Príklad 2:

Pre väčšie n možno binomické rozdelenie úspešne aproximovať Poissonovým. Najmä ak chceme počítať pravdepodobnosť len pre niekoľko hodnôt *k*, použitie vzorca pre Poissonovo rozdelenie je jednoduchšie.

Zvoľme L=n*p. Hodnoty Poissonovho rozdelenia získame v duchu predošlých úvah opäť rekurentným spôsobom:

```
L=n*p;
pp(1) = exp(-L);
for k = 1:n, pp(k+1) = pp(k)*L/k; end
sum(pp)
hold on, bar(0:n,pp), colormap([0 0.5 0])
```

Ak si zväčšíme získaný obrázok, v detailných náhľadoch uvidíme, ako sa k sebe blížia obe rozdelenia: