

Teorija informacije – 1. labos - 2007./08.

Rješenja

1. Funkcija ti

1.1, 1.2, 1.3 -> s ovim ne bi trebalo bit problema.

2. Markovljev izvor nultog reda

2.1 $q = \text{rand}(1,3)$ (naredba $\text{rand}(m,n)$ stvara random matricu $m \times n$ s elementima iz $[0,1]$)

2.2 $q = q / (\text{sum}(q(1,1:3)))$ (matricu q normiramo na matricu vjerojatnosti tako da ju (tj. svaki njen element) podijelimo sa sumom svih njenih elemenata. sada im je ukupni zbroj jednak jedinici)

2.3 $z = [q(1,1:3); q(1,1:3); q(1,1:3)]$ (u novoj matrici prijelaznih vjerojatnosti imamo tri retka od kojih je svaki isti kao i početna matrica q . za pojasnjenja vidi str 2, (4), al radi se samo o tome da u M.I. nultog reda iduci generirani simbol ne ovisi o prethodnom)

$w = \text{ti}(1000,3,z)$ (generiramo slucajni slijed duljine 1000 s 3 razlicita simbola i matricom prijelaza z)

2.4 $\text{mean}(w)$ (srednja vrijednost elemenata niza w)
 $\text{std}(w)$ (standardna devijacija niza w)

2.5 $\text{hist}(w,3)$ (ovom naredbom ce komp nacrtat histogram a vi ga samo precrtate u pripremu)
 $p = [\text{size}(\text{find}(w==1),1)/1000 \text{ size}(\text{find}(w==2),1)/1000 \text{ size}(\text{find}(w==3),1)/1000]$ ($\text{find}(w==1)$ napravi matricu s jednim stupcem i onoliko redaka koliko se puta 1 pojavljuje u nizu w . size izbroji koliko je tih redaka te se taj broj podijeli s ukupnim brojem elemenata w)

2.6 $-(q(1,1) \cdot \log_2(q(1,1)) + q(1,2) \cdot \log_2(q(1,2)) + q(1,3) \cdot \log_2(q(1,3)))$ (obicna formula za entropiju s tim da vjerojatnosti pojave pojedinog elementa citamo iz matrice q)

3. Markovljev izvor prvog reda

3.1 $x = \text{ti}(1000,4)$ (isto kao 2.3 s tim da se trazi da matricu q ne zadajemo)

3.2 $\text{hist}(x,4)$ (crta histogram, pa onda ili mozete vjerojatnosti ocitati iz histograma ako malo zumirate (procitani broj/1000) i rucno unjeti u matricu p ili napisete naredbu: $p = [\text{size}(\text{find}(w==1))/1000 \text{ size}(\text{find}(w==2))/1000 \text{ size}(\text{find}(w==3))/1000 \text{ size}(\text{find}(w==4))/1000]$, kak smo to napravili u 2.5)

3.3 ne znam kako su mislili da ovo napravimo al ja ne znam puno sretniji nacin nego napisati neku .m funkciju koja ce te vjerojatnosti racunat iz vektora x . pa ako ima ko neku sretniju ideju nek ju kaze slobodno...

3. Da bi pozvali funkciju ti za izvorište **nultog reda**, nakon normiranja vektora q trebamo ga proširiti u matricu kojoj su svi retci jednaki (navedeno zašto u uputi). To se radi dvostrukom for petljom. Takvu matricu ubacujemo u funkciju ti pa poziv izgleda ovako:

$w = \text{ti}(1000,3,q)$

4. $m_i = \text{mean}(w)$
 $\text{dev} = \text{std}(w)$

5. $\text{hist}(w)$
 $p_1 = \text{size}(\text{find}(w==1),1)/1000$
 $p_2 = \text{size}(\text{find}(w==2),1)/1000$
 $p_3 = \text{size}(\text{find}(w==3),1)/1000$

ako se zbroje za sva tri slučaja, dobije se 1.

6. $H = 0;$
for $i = 1:3$
 $H = H - q(i) \cdot \log_2(q(i));$
end
 H

3.) Markovljev izvor prvog reda

1. Poziv funkcije je sad $s=ti(1000,4)$

2. `hist(s)`

Vjerojatnosti se dobiju naredbom $n=hist(s,4)$

$p=$

391 171 235 203

Ispred svake vrijednosti treba dodati decimalnu točku.

$p=p/1000$

3. Treba odrediti matricu uvjetnih vjerojatnosti, a to se izvodi pomoću nove m.funkcije :

```
function q=uvj(s)
for i=1:4
for j=1:4
q(i,j)=0;
for k=2:1000
if((s(k)==j) & (s(k-1)==i))
q(i,j)=q(i,j)+1;
end
end
q(i,j)=q(i,j)./size(find(s==i),1);
end
end
```

4. $p*q=p$ (Zbilja se dobije isti rezultat)

5. kao i u prvom primjeru

6. nisam siguran, ali mislim da je

$p'=[q_{11} \ q_{12} \ q_{13}]$

$p_1=q(1,:)$

$p''=[p_{11} \ p_{12} \ p_{13}] * q$

$p_{11}=p_1*q$

7. Vjerojatno nacrtati četiri stanja i vjerojatnosti iz matrice uvjetnih vjerojatnosti q kao u uvodu

8. Ovdje se računa entropija no prvo moramo izračunati vjerojatnosti izlaza $[p(y_j)]$:

```
for i=1:4
for j=1:4
pq(i,j)=p(i)*q(i,j);
end
end
py=sum(pq)
```

Nastavak je kao u prvom primjeru osim što umjesto q uvrštavamo py .

3.4 p (ispišemo matricu p)

$p*q$ (ispišemo matricu $p*q$ i pogledamo jel to isto)

3.5 `mean(x)`

`std(x)`

3.6 $p_1=[1 \ 0 \ 0 \ 0]*q$ (dakle zadano nam je da je izvor u stanju 1 pa na temelju toga određujemo vjerojatnosti odabira pojedinog idućeg stanja)

$p_2=p_1*q$ (analogno prethodnom računamo vjerojatnosti u drugom koraku, detaljnije na str. 4)

3.7 graf prijelaza sastoji se od 4 cvora od kojih je svaki povezan sa svakim drugim. vjerojatnost prelaska iz nekog u neki cvor citate iz matrice q (nesretni zadatak 3.3:))

3.8

Cini mi se da jos nigdje nije objasnjeno, pa evo mog vidjenja funkcije ti(...) za generiranje niza slucajnih brojeva.

Rad funkcije ako nije zadana slucajna matrica q:

```
function z = ti(x, y, q)
if (nargin<3)
q = rand(y);
end
```

```
-- zadaju se parametri funkcije i, ako nije zadana, slucajna matrica q generira se pomocu funkcije rand()
-- svi elementi matrice su u intervalu [0, 1>
-- sad pretpostavimo da se generirala sljedeća matrica (u zadatku je dimenzija 4x4, ali za objasnjavanje principa rada funkcije nije bitno)
```

```
mat1
[0.2, 0.3, 0.5
0.7, 0.1, 0.2
0.5, 0.4, 0.1]
```

```
-- matrica ima ovo znacenje: element q(i, j) je vjerojatnost da Markovljev izvor nakon sto je generirao element i generira element j
```

```
--funkcija radi sljedeće:
for i=1:y
q(i, :) = cumsum(q(i, :));
q(i, :) = q(i, :)/q(i, y);
end
```

```
--u ovom dijelu funkcija svodi slucajnu matricu na oblik koji joj je pogodan za generiranje slucajnog vektora z
--prvo svaki element q(i, j) zamjenjuje zbrojem elemenata u i-tom retku koji su ispred tog elementa ukljucujuci i doticni element, a zatim dijeli cijeli taj redak matrice zadnjim elementom u retku
-- matrica je sad u ovom obliku
```

```
mat2
[0.2, 0.5, 1
0.7, 0.8, 1
0.5, 0.9, 1]
```

```
--kad svedemo matricu na ovaj oblik, mozemo pomocu funkcije rand() generirati niz slucajnih brojeva, ali moramo paziti koji smo simbol zadnji generirali (jer se bavimo izvorom prvog reda)
```

```
--princip je sljedeći:
u mat1 imamo vjerojatnosti generiranja elementa j ako je prehodno generiran element i
npr. ako je generirani element 1, onda su vjerojatnosti da je iduci element [1, 2, 3] jednake [0.2, 0.3, 0.5]
-u mat2 smo zapisali te vjerojatnosti kao [0.2, 0.5, 1] pri čemu nam je bitna razlika između susjednih elemenata
-pozovemo funkciju rand() koja vraća neku vrijednost x u intervalu [0, 1>
```

```
sad imamo tri slucaja:
- ako je x < 0.2, generiramo broj 1
- ako je 0.5 > x > 0.2 , generiramo broj 2
- ako je x > 0.5, generiramo broj 3
```

vjerojatnosti generiranja brojeva su <0.2 - 0, 0.5 - 0.2, 1 - 0.5> sto odgovara vjerojatnostima pojavljivanja brojeva u mat1

u funkciji:

```
z = zeros(x, 1);
```

```
--generiramo vektor stupac od x elemenata pri čemu je prvi element 1, a svi ostali su jednaki nuli
--u vektor z spremamo generirane simbole, a ujedno nam služi i za određivanje prethodnog simbola (jer o njemu ovise vjerojatnosti generiranja sljedećeg)
```

```
z(1) = 1;
--prvi element u vektoru x je 1 pa smo time pretpostavili da je Markovljev izvor na pocetku generirao simbol 1 (mogli smo odabrati i neki drugi)
```

```
for i=2:x // dok ne popunimo vektor z
s = rand(1) // generiramo slucajni broj
s1 = find(q(z(i-1), :)>s); // trazimo indekse brojeva u retku z(i-1) mat2 koji su veci od generiranog broja
```

```
z(i) = s1(1); /spremimo prvi index u vektor z
end
```

```
npr. z = [1;3;3;0;0;0;0;0;0;0], i = 4 //u stanju 3 smo (prethodno je generiran simbol 3)
buduci da smo u stanju 3, vjerojatnosti generiranja pojedinih simbola su: 0.5, 0.4, 0.1 ili drugacije zapisano 0.5, 0.9, 1
```

```
s = rand(1) - npr. 0.83
s1 = find(q(z(i-1), :)>s); //-> s1 = find(q(3, :)>0.83) ->s1 = [2 3]
z(i) = s1(1) ->z(4) = 2 //stavljamo generirani element u vektor z
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% TINF - Markovljeva izvorišta %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% 1.1 - funkcija "ti" - spremi kao .m
file
% function z=ti(x,y,q)
% if (nargin<3)
% q=rand(y);
% end
% for i=1:y
% q(i,:)=cumsum(q(i,:));
% q(i,:)=q(i,:)/q(i,y);
% end
% z=zeros(x,1);
% z(1)=1;
% for i=2:x% s=rand(1);
% s1=find(q(z(i-1),:)>s);
% z(i)=s1(1);
% end

clear
clc

% 2.1
q=rand(1,3);
disp(['q=[' num2str(q) '].']);
pause
% 2.2
q=q./sum(q);
disp(['q=[' num2str(q) '].']);
pause
% 2.3
z=ti(1000,3,[q;q;q]);
% 2.4
disp(['Srednja vrijednost je: '
num2str(mean(z)) '.']);
disp(['Standardna devijacija je: '
num2str(std(z)) '.']);
pause
% 2.5
hist(z,3);
pause
for i=1:3
disp(['p(' num2str(i) ')='
num2str(size(find(z==i),1)/1000) '.']);
end
clear i
pause
% 2.6
disp(['H(izv.)=' num2str(-sum(q.*log2(q)))
' [bita/simbolu]']);
pause

clear
clc

% 3.1
z=ti(1000,4);
% 3.2
p=hist(z,4)./1000;
disp(['p=[' num2str(p) ']]);
pause
% 3.3
for i=1:4
for j=1:4
q(i,j)=0;
for k=2:1000
if ((z(k)==j) & (z(k-1)==i))
q(i,j)=q(i,j)+1;
end
end
q(i,j)=q(i,j)./size(find(z==i),1);
end
end
clear i j k
disp(['q=p(xj|yi)=']);
disp(q);
disp('].');
pause
% 3.4
disp(['pxq=[' num2str(p*q) '].']);
pause
% 3.5
disp(['Srednja vrijednost je: '
num2str(mean(z)) '.']);
disp(['Standardna devijacija je: '
num2str(std(z)) '.']);
pause
% 3.6
disp(['p'=[' num2str(q(1,:)) '].']); %
formula (7) iz pripreme
disp(['p'=[' num2str(q(1,:)*q) '].']); %
formula (8) iz pripreme
pause
% 3.8
for i=1:4
for j=1:4
pq(i,j)=p(i)*q(i,j);
end
end
py=sum(pq);
clear i j
disp(['H(izv.)=' num2str(-
sum(py.*log2(py))) ' [bita/simbolu]']);

clear
clc

```