

Gheorghe M.Panaitescu

GHID DE LUCRĂRI

la disciplina

FIABILITATE SI DIAGNOZĂ

**Universitatea “Petrol-Gaze” Ploiesti
Catedra Automatică si calculatoare
2007**

INTRODUCERE

Prezentul **Ghid** este destinat sustinerii unor aplicatii la cursul de **Fiabilitate si diagnoză** din programul de pregătire al anului V la specializările Automatică si informatică industrială si Electronică. **Ghidul** contine sase lucrări de dificultăți variate. Unele, mai simple se intind pe durata unei singure sedinte, altele, mai complexe pot dura până la trei sedinte consacrate lucrărilor aplicative.

La una din lucrări, uzual la Lucrarea 3, se cere un **Referat tehnic** care se dorește o simulare a unor rapoarte explicative relativ la lucrări pe care în calitate de inginer, viitorul absolvent le va avea de realizat întru satisfacerea unor comenzi ale unor potentiali solicitanti. Se sugerează imediat o structură a unui asemenea **Referat**.

GHID DE REDACTARE A REFERATELOR TEHNICE

Referatele tehnice asupra unor lucrări efectuate în orele de aplicatii au ca prim scop o verificare partială a cunostintelor.

Al doilea scop urmărit, care nu este mai puțin important constă în testarea capacității autorului de a aseza în pagină, în stil **concis si clar, enuntul, scopul si solutia** problemei care constituie obiectul lucrării. Este, sub acest aspect, o simularea a cazurilor în care un inginer este pus în situatia de a elabora un raport scris asupra lucrărilor sale. Punctualitatea depunerii raportului este, de asemenea, apreciată.

Cu aceste precizări, referatele trebuie să contină:

- formularea obiectivului lucrării
- un minim suport teoretic cu eventuale trimiteri bibliografice
- în ce constă solutia si mijloacele prin care a fost obtinută
- răspunsuri sustinute de explicatii la chestiunile ridicate în ghidul scris al lucrării
- concluzii

În plus:

- Raportul trebuie să se întindă pe cel mult 5-6 pagini
- Raportul trebuie să fie individual; rapoartele identice, diferite doar prin semnătură vor fi apreciate negativ
- Raportul trebuie predat exact la data indicată verbal de conducătorul lucrărilor

Lucrarea selectată pentru Referat, termenul de predare și alte posibile detalii se stabilesc în una din primele sedințe de aplicații.

C U P R I N S U L

LUCRAREA 1. Generarea de numere aleatoare, verificarea experimentală a unor legi de repartitie	7
LUCRAREA 2. Legi de repartitie utilizate in teoria fiabilității sistemelor	15
LUCRAREA 3. Fiabilitatea sistemelor cu schimbare/reînnoire	23
LUCRAREA 4. Fiabilitatea sistemelor <i>software</i>	31
LUCRAREA 5. Analiza componentelor principale (ACP) în diagnoză	37
LUCRAREA 6. Rețele neuronale artificiale în diagnoză	43

LUCRAREA 1

GENERAREA DE NUMERE ALEATOARE, VERIFICAREA EXPERIMENTALĂ A UNOR LEGI DE REPARTITIE

1. Obiectivele lucrării

- Studiul generatoarelor de numere aleatoare de bibliotecă (PASCAL, C++, BASIC, MATLAB etc.)
- Studiul altor generatoare de numere aleatoare

2. Aparatură si suport documentar

- Calculatoare PC în configurație obișnuită
- Prezentul *Ghid de lucrări*
- Notele de curs la disciplina *Fiabilitate si diagnoză*
- Manuale de programare, documentarea *on-line (Help)*

3. Breviar teoretic

Bibliotecile limbajelor de programare cele mai utilizate contin cel puțin o functie generatoare de numere aleatoare repartizate uniform pe o multime precizată. În PASCAL de pildă, functia *random* fără argument generează numere aleatoare reale, uniform repartizate pe intervalul (0,1). Aceasta înseamnă că numerele subunitare pozitive sunt produse la întâmplare și au șanse egale de apariție. Utilizarea prealabilă a procedurii *randomize* are un efect de initializare și asigură un start totdeauna diferit în generarea unei secvențe de astfel de numere aleatoare.

Aceeași functie, dar cu un argument de tip word, *random(w)*, generează numere aleatoare de tipul argumentului, cuprinse între 0 și $w - 1$.

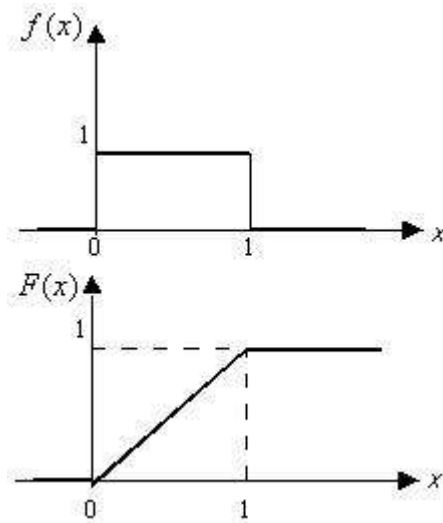
Pentru cazul (pseudo)continuu al funcției *random* fără argument se poate figura funcția densitate de repartiție și funcția de repartiție ceea ce s-a și făcut în graficele alăturate.

Cu generatorul de numere aleatoare *random* sau cu generatoarele similare din alte limbaje se pot genera numere aleatoare repartizate după legi diferite de cea uniformă. Pentru aceasta se pot utiliza metode derivate pe cale analitică sau o metodă directă care are în vedere funcția de repartiție a variabilei de generat.

Legea de repartitie normală normată (de medie nulă și de dispersie 1) este legată de legea de repartitie uniformă prin una sau alta dintre relațiile următoare

$$u_1 = \sqrt{-2 \ln x_1} \cos(2\pi x_2)$$

$$u_2 = \sqrt{-2 \ln x_1} \sin(2\pi x_2)$$



în care x_1 și x_2 sunt două numere aleatoare independente cu repartitie uniformă în intervalul $(0,1)$. Este un exemplu de generare analitică a unor numere aleatoare care au altă lege de repartitie, diferită de cea uniformă.

Un alt exemplu este cel al generării de numere aleatoare uniform repartizate pe un interval oarecare finit (a,b) , $a < b$. Trecerea la noua variabilă se realizează prin mijlocirea relației

$$u = a + (b - a)x$$

cu x generat de funcția de bibliotecă *random*. Variabila u este uniform repartizată pe intervalul finit specificat.

Variantele analitice de generare a unor numere aleatoare repartizate conform unei legi particulare nu sunt totdeauna posibile și accesibile. Modul de generare alternativ este descris în continuare.

Fiind dată funcția de repartiție $F(u)$ sau funcția densitate de repartiție $f(u)$ – caz în care se poate calcula $F(u)$ – pentru o variabilă care urmează o anumită lege de repartiție, se generează valori x uniform repartizate pe intervalul $(0,1)$ cu ajutorul funcției de bibliotecă *random* (sau similare din alte limbaje de programare). Se calculează de fiecare dată $u = F^{-1}(x)$, unde $F^{-1}(\cdot)$ este inversa funcției de repartiție a variabilei u , care trebuie generată. Funcția de repartiție este totdeauna o funcție monotonă, deci este inversabilă pentru orice $x \in (0,1)$. Intervalul $(0,1)$ este mulțimea de valori comună tuturor funcțiilor de repartiție. Variabila aleatoare u este cu siguranță repartizată conform legii date de funcția $F(u)$ (sau de $f(u)$).

Uneori, pe cale experimentală sau prin generare conform algoritmilor expuși mai devreme, se acumulează o listă de valori ale unei variabile aleatoare. În ambele cazuri, dar mai ales în cazul unei liste de sursă experimentală, se poate pune problema verificării legii de repartiție statistică pe care o urmează variabila aleatoare observată/generată. Între metodele de verificare a ilustrării prin experiment a unei legi de repartiție sau de verificare a adecvării unui model teoretic la realitatea sondată prin

valorile din lista menționată, cea mai eficace pare a fi cea bazată pe variabila χ^2 . Variabila χ^2 este o variabilă aleatoare care rezultă din însumarea pătratelor unor variabile aleatoare z normale normate (de medie nulă și de dispersie 1). Numărul variabilelor z reprezintă numărul gradelor de libertate ale variabilei χ^2 . Dacă lista de valori observate/generate este t_i ($i = 1, 2, \dots, n$), algoritmic se parcurg următoarele etape:

- a) Se sortează valorile t_i pe m intervale succesive, adiacente, de lărgime convenabilă, care formează o așa-numită partiție a intervalului care conține valorile t_i
- b) Se calculează frecvențele n_k și frecvențele relative n_k/n pentru fiecare interval I_k ($k = 1, 2, \dots, m$)
- c) Se calculează pentru aceleași intervale probabilitățile conform den-

sității de repartiție presupusă pentru variabila t , $p_k = \int_{I_k} f(t) dt$

- d) Se calculează valoarea $\sum_{k=1}^m \frac{(n_k - np_k)^2}{np_k}$, care este o variabilă χ^2 cu m grade de libertate
- e) Se compară valoarea χ^2 calculată cu date tabelare corespunzătoare gradelor de libertate specifice și unui nivel de semnificație ales, uzual de 95%. În tabelul alăturat sunt date câteva valori corespunzătoare nivelului de semnificație de 95%, pentru grade de libertate f diferite. Depășirea valorilor tabelare conduce la concluzia că modelul admis

nu este adecvat materialului experimental sau, invers, dacă evaluarea bazată pe lista experimentală produce o valoare inferioară față de tabele, lista de valori provine dintr-o populație statistică guvernată de legea de repartiție presupusă și supusă verificării.

f	5	6	7	8
χ^2	11.070	12.592	14.067	15.507
f	9	10	15	20
χ^2	16.919	18.307	24.996	31.410

4. Modul de lucru

A. Se scrie un program capabil de a genera o secvență de numere aleatoare uniform repartizate în intervalul (0,1). Se împarte intervalul în subintervale acoperitoare, se face cu alte cuvinte o partiție a intervalului (0,1). Se aplică algoritmul de verificare experimentală a unei legi de repartiție prezentat mai sus.

B. Se scrie un program (sau se completează cel de la punctul A) care să genereze pe cale analitică numere aleatoare repartizate normal. Pe o partiție potrivită a axei reale se efectuează o verificare urmând algoritmul de mai sus.

C. Se construiește o funcție de repartiție (de pildă din segmente) și se generează numere aleatoare care să urmeze legea de repartiție propusă prin construcție (a se vedea algoritmul de generare prezentat în **Breviarul**

teoretic). Generatorul poate fi un program separat sau o secvență în programul anterior.

Programele pot avea, se recomandă chiar a avea secvențe grafice.

5. Chestiuni de studiu

- Tendința de uniformizare a frecvențelor absolute și relative ale aparițiilor numerelor generate cu funcția *random* (sau similară) pe măsură ce numărul de generări crește
- Tendința de apropiere a frecvențelor relative ale numerelor aleatoare normal repartizate generate analitic de probabilitățile calculate din funcția densitate de repartiție teoretică, pe măsură ce numărul de generări crește
- Verificarea cu criteriul χ^2 a impresiei vizuale, numerice și/sau grafice, de apropiere între frecvențele relative și probabilități

6. Glosar

Frecvență absolută: rezultatul numărării valorilor experimentale care aparțin unui subinterval precizat

Frecvență relativă: raportul frecvenței absolute pentru un subinterval la suma tuturor frecvențelor absolute asociate tuturor subintervalurilor definite

Histogramă: Echivalentul grafic (prin segmente verticale sau prin dreptunghiuri) al unui tabel al frecvențelor absolute/relative

Poligon al frecvențelor cumulate: Graficul echivalent unei funcții de repartiție

random: funcție din biblioteca PASCAL, generatoare de numere uniforme repartizate în intervalul (0,1) când este apelată fără argument, sau de numere uniforme repartizate pe mulțimea finită $\{0, 1, \dots, w - 1\}$ când este apelată cu argumentul w , întreg de tipul *word*;

randomize: procedură care initializează totdeauna alt fel generatorul *random*.

LUCRAREA 2

LEGI DE REPARTITIE UTILIZATE IN TEORIA FIABILITĂȚII SISTEMELOR

1. Obiectivele lucrării

- Cunoasterea unor legi de repartitie a duratei de viață a sistemelor, frecvent utilizate în studiile de fiabilitate
- Verificarea adecvării unei legi de repartitie la un set de observatii experimentale asupra defectării sistemelor

2. Aparatură si suport documentar

- Calculatoare PC în configurație obisnuită
- Prezentul *Ghid de lucrări*

- Notele de curs de la disciplina *Fiabilitatea si diagnoza* si orice alte surse bibliografice din domeniu
- Manuale de programare, documentarea *on-line (Help)*

3. Breviar teoretic

Pentru sistemele fără reînnoire, adică acele sisteme care odată defecte sunt iremediabil defecte, durata lor de viață este o variabilă aleatoare. O variabilă aleatoare este complet definită de funcția ei de repartiție. Durata de viață este o variabilă aleatoare de tip continuu, prin urmare funcția de repartiție este o funcție continuă și derivabilă în raport cu timpul până la prima (și ultima) defectare. În cazul continuității variabilei, densitatea ei de repartiție este de asemenea capabilă să o descrie complet. Tabelul dat în aceste pagini conține un număr de densități de repartiție a duratei de viață a sistemelor, densități foarte frecvent utilizate și confirmate de practică.

4. Chestiuni de studiu

- Verificarea numerică a calității de densități de repartiție $\int_{-\infty}^{+\infty} f(t)dt = 1$ a funcțiilor din tabel;
- Verificarea prin calcul a expresiilor pentru medii și dispersii;

Legi de repartitie utilizate în teoria fiabilității sistemelor

Nume	$f(t)$	Media	Dispersia
Gama	$\frac{\beta (\beta t)^{\alpha-1} e^{-\beta t}}{\Gamma(\alpha)}$	$\frac{\alpha}{\beta}$	$\frac{\alpha}{\beta^2}$
Weibull	$\frac{\beta t^{\beta-1}}{\alpha} e^{-\frac{t^\beta}{\alpha}}$	$\alpha^{\frac{1}{\beta}} \Gamma\left(1 + \frac{1}{\beta}\right)$	$\alpha^{\frac{2}{\beta}} \left[\Gamma\left(1 + \frac{2}{\beta}\right) - \Gamma^2\left(1 + \frac{1}{\beta}\right) \right]$
Normală	$\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{t-\mu}{\sigma}\right)^2}$	μ	σ^2
Lognormală	$\frac{1}{\sqrt{2\pi}\sigma t} e^{-\frac{1}{2}\left(\frac{\ln t - \mu}{\sigma}\right)^2}$	$e^{\mu + \frac{\sigma^2}{2}}$	$e^{2\mu + \sigma^2} (e^{\sigma^2} - 1)$
A valorii extreme (Gumbel)	$e^{\frac{t-\mu}{\eta}} e^{-e^{\frac{t-\mu}{\eta}}} \left(\eta e^{-e^{\frac{t-\mu}{\eta}}} \right)$		
Rayleigh	$\frac{2}{\omega^2} t e^{-\frac{t^2}{\omega^2}}$	$\omega \Gamma\left(\frac{3}{2}\right)$	$\omega^2 \left[1 - \Gamma^2\left(\frac{3}{2}\right) \right]$
Rayleigh generalizat	$\frac{2\theta^{k+1}}{\Gamma(k+1)} t^{2k+1} e^{-\theta t^2}$	$\frac{\Gamma\left(k + \frac{3}{2}\right)}{\Gamma(k+1)} \frac{1}{\sqrt{\theta}}$	$\left[k + 1 - \frac{\Gamma^2\left(k + \frac{3}{2}\right)}{\Gamma^2(k+1)} \right] \frac{1}{\theta}$
Alfa	$\frac{\beta}{t^2 \sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\beta}{t} - \alpha\right)^2}$	$\frac{\beta}{\alpha} \left(1 + \frac{1}{\alpha^2} \right)$	$\frac{\beta^2}{\alpha^4} \left(1 + \frac{8}{\alpha^2} \right)$
Putere	$\delta b^{-\delta} t^{\delta-1} \quad t \in (0, b)$	$\frac{\delta b}{\delta + 1}$	$\frac{\delta b^2}{(\delta + 2)(\delta + 1)^2}$
Birnbaum-Saunders	$\frac{1}{2\sqrt{2\pi}\alpha t} \cdot \left(\sqrt{\frac{t}{\beta}} + \sqrt{\frac{\beta}{t}} \right) \cdot e^{-\frac{1}{2\alpha^2} \left(\frac{t}{\beta} + \frac{\beta}{t} - 2 \right)}$	$\beta \left(1 + \frac{\alpha^2}{2} \right)$	$(\alpha\beta)^2 \left(1 + \frac{5\alpha^2}{4} \right)$

- Reprezentarea grafică a funcțiilor din tabel, pentru valori ale parametrilor continuti, la liberă alegere;
- Reprezentarea grafică a funcțiilor de repartitie, a funcțiilor de fiabilitate si a ratei de defectare pentru fiecare din funcțiile din tabel;
- Simularea unui proces de observare a n sisteme identice sub aspectul fiabilității lor.

5. Modul de lucru

Pentru verificarea faptului că funcțiile din tabel sunt densități de repartitie, precum si pentru reprezentarea lor grafică se cere a se realiza scurte programe într-un limbaj de programare convenabil, indiferent că unele sugestii formulate în continuare sunt cu referire la limbajul *Pascal* sau la *Matlab*.

Pentru unele functii se sugerează scriptul *Matlab* care urmează:

```
% Curbe f(t), F(t) si h(t) în cazul repartitiei Weibull
% b<1 => rata de defectare scăzătoare
% b>1 => rata de defectare crescătoare
% b=1 => rata de defectare constantă
clear x y z u
a=0.02;b=2.2;
i=0;c=0;
while c<0.99
```

```

    i=i+1;
    x(i)=i;
    z(i)=weibcdf(x(i),a,b);c=z(i);
    y(i)=weibpdf(x(i),a,b);
    u(i)=y(i)/(1.0-z(i));
end
subplot(1,3,1)
plot(x,y);grid on;title('f(t)')
subplot(1,3,2)
plot(x,z);grid on;title('F(t)')
subplot(1,3,3)
plot(x,u);grid on;title('z(t)')

```

Funcțiile `weibpdf` și `weibcdf`, ca și altele câteva din tabel, sunt implementate ca funcții de bibliotecă în pachetul `stats`.

Pentru simularea unui proces de observare, se selectează o lege de repartiție convenabilă, și se generează n durate de viață la întâmplare, care să se distribuie statistic conform legii de repartiție aleasă. Procedura care trebuie urmată este:

- Calculați funcția de repartiție $F(t) = \int_{-\infty}^t f(t)dt$
- Calculați inversa acesteia F^{-1} și prin generarea de numere aleatoare r_i uniform repartizate pe intervalul $(0,1)$ (cu funcția Pascal *random* fără argument, de pildă) rezolvați de n ori ecuația $F^{-1} = r_i$ ($i = 1,2,\dots,n$) în care r_i este numărul aleator generat în “experiența” i

- Sortati solutiile t_i pe m intervale succesive, adiacente, de lărgime convenabilă
- Calculati frecventele n_k si frecventele relative n_k/n pentru fiecare interval $I_k (k = 1, 2, \dots, m)$
- Calculati pentru aceleasi intervale probabilitățile conform densității de

repartitie, $p_k = \int_{I_k} f(t) dt$.

Desi este de asteptat ca frecventele relative rezultate “experimental” să confirme legea de repartitie teoretică aleasă, faceti o verificare a reprezentativității legii de repartitie teoretice pentru “experimentul” observat, prin intermediul criteriului statistic χ^2 . Calculati valoarea

criteriului cu relatia $\chi^2 = \sum_{k=1}^m \frac{(n_k - np_k)^2}{np_k}$. Se obtine o valoare χ^2 cu m grade de libertate. Valoarea obtinută se compară apoi cu date tabelare pentru nivelul de semnificatie de 95%.

f	5	6	7	8
χ^2	11.070	12.592	14.067	15.507
f	9	10	15	20
χ^2	16.919	18.307	24.996	31.410

Modificarea arbitrară a unor observatii poate fi instructivă referitor la sensibilitatea criteriului statistic χ^2 .

6. Glosar de termeni

frecvență (absolută) – numărul de observații care întrunesc o anumită condiție, de pildă apartenența la un interval

frecvență relativă – numărul subunitar care se obține prin împărțirea frecvenței unor observații grupate după un anumit criteriu la numărul total de observații

funcție de repartiție – funcție care exprimă probabilitatea ca o variabilă aleatoare să ia o valoare inferioară unei valori precizate

densitate de repartiție (de probabilitate) – funcția derivată a funcției de repartiție, dacă această derivată există

funcția euleriană de specie I, $\Gamma(x)$ – funcție definită de integrala

improprie cu parametru $\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt$, convergentă pentru orice $x >$

0; funcția are proprietatea importantă $\Gamma(x + 1) = x \Gamma(x)$, ceea ce face posibil calculul funcției pentru orice valoare a lui x dacă sunt cunoscute valorile pe un interval de lungime egală cu unitatea. În particular, pentru x întreg funcția ia valoarea $\Gamma(x + 1) = x!$.

LUCRAREA 3

FIABILITATEA SISTEMELOR CU SCHIMBARE/REÎNNOIRE

1. Obiectivele lucrării

- Studiul fiabilității unor sisteme cu schimbare/reînnoire pe exemple simulate
- Evaluarea unor parametri de natură economică legați de siguranța în funcționare a sistemelor cu reînnoire

2. Aparatură și suport documentar

- Calculatoare PC în configurație obișnuită
- Prezentul *Ghid de lucrări*
- Notele de curs de la disciplina *Fiabilitate și diagnoză* și orice alte surse bibliografice din domeniu

- Manuale de programare, documentarea *on-line* (*Help*)

3. Breviar teoretic

Un sistem cu reînnoire este caracterizat de momentele de reînnoire t_1, t_2, \dots, t_n și de intervalele între reînnoiri X_1, X_2, \dots, X_n . Numărul N_t de reînnoiri petrecute într-un interval precizat $(0, t)$ se constituie într-un proces aleator discret. În ceea ce privește relația între variabilele aleatoare $X_i, i = 1, 2, \dots, n$, independența statistică reciprocă pentru indici diferiți este o ipoteză acceptată ca rațională. Acest fapt permite tratarea în fiecare interval a aceluiași indicator de fiabilitate ca în oricare din intervalele premergătoare, independenți unii de alții.

Cu notația $R_i(x)$ pentru funcția de fiabilitate pe intervalul (t_{i-1}, t_i) , reînnoirile se clasifică după relația între funcțiile de fiabilitate pe diferite intervale. Reînnoire propriu-zisă este o reînnoire care aduce sistemul de fiecare dată în starea de dinaintea defectării, adică $R_i(x) = R(x)$ pentru orice i . Un asemenea proces se numește proces de reînnoire simplu.

Între cazurile multiple posibile se disting reînnoirile pozitive și reînnoirile negative cu $R_1(x) > R_2(x) > \dots > R_n(x)$, respectiv cu $R_1(x) < R_2(x) < \dots < R_n(x)$.

Dacă se notează cu T_r variabila aleatoare definită ca durata scursă până la reînnoirea a r -a și cu N_t numărul de reînnoiri până la momentul t , atunci,

în cazul unui sistem cu reînnoire propriu-zisă, are loc egalitatea de probabilități

$$P(N_t \geq r) = P(T_r < t)$$

cu alte cuvinte, numărul de reînnoiri produse în intervalul $(0, t)$ este mai mare decât r dacă și numai dacă durata T_r scursă până la reînnoirea cu numărul r este inferioară lui t .

Se notează cu $K_r(t)$ funcția de repartiție a duratei T_r și cu $k_r(t)$ densitatea ei de probabilitate. Procesul aleator N_t poate fi exprimat cu ajutorul acestor funcții

$$P(N_t = r) = P(N_t \geq r) - P(N_t \geq r + 1) = K_r(t) - K_{r+1}(t)$$

cu $r = 1, 2, \dots$ și $K_0(t) = 1$.

Variabila $T_r = X_1 + X_2 + \dots + X_r$ are ca densitate de repartiție, totuna cu densitatea de probabilitate, o convoluție multiplă de r factori identici

$$k_r(t) = f(t) \otimes f(t) \otimes \dots \otimes f(t)$$

care are transformata Laplace (scrisă pretutindeni ca o funcție de s cu același nume/semn grafic marcat cu un asterisc)

$$k_r^*(s) = [f^*(s)]^r$$

Conform unei reguli binecunoscute, funcția de repartiție are transformata Laplace

$$K_r^*(s) = (1/s)[f^*(s)]^r$$

Funcția de reînnoire este

$$H(t) = \sum_{r=1}^{\infty} rP(N_t = r) = \sum_{r=1}^{\infty} r[K_r(t) - K_{r+1}(t)] = \sum_{r=1}^{\infty} K_r(t)$$

si densitatea de reînnoire este

$$h(t) = \frac{dH(t)}{dt} = \sum_{r=1}^{\infty} k_r(t)$$

În domeniul Laplace, pentru reînnoirea simplă

$$h^*(s) = \sum_{r=1}^{\infty} [f^*(s)]^r = \frac{f^*(s)}{1 - f^*(s)}$$

si

$$H^*(s) = \frac{f^*(s)}{s[1 - f^*(s)]}$$

4. Modul de lucru

Se propune pentru analiză un sistem cu reînnoire propriu-zisă, cu alte cuvinte un sistem care la fiecare defectare este readus la parametrii initiali, astfel încât functia de fiabilitate este $R_i(x) = R(x)$, aceiasi pentru orice indice i , adică în orice interval X_i dintre două defectări succesive.

Pentru sistemul cu reînnoire, pentru intervalul premergător primei reînnoiri, se preia din tabelul studiat în **Lucrarea 2** una din legile de repartitie a duratelor de viață, de pildă legea Weibull

$$f(t) = \frac{\beta}{\alpha} t^{\beta-1} e^{-\frac{t^\beta}{\alpha}}$$

Se dau valori particulare parametrilor distributiei retinute; pentru densitatea de repartitie Weibull se consideră potrivite valorile $\alpha = 800.000$, $\beta = 2,5 + p/100$, cu p numărul de ordine al studentului în catalogul grupei.

Cu sau fără referire la lucrarea precedentă, se cere elaborarea unui program care să modeleze sistemul cu reînnoire propus, pe o durată indefinită.

Problema se poate rezolva pe mai multe căi. Se propun mai jos câteva căi de abordare posibile.

Posibilitatea *primă* este dată de transformarea Laplace conform secțiunii **Breviar teoretic** de mai sus. Calculul efectiv ar putea fi dificil pentru motivul simplu că funcția densitate de repartitie $f(t)$ nu este printre cele uzuale, care să poată fi găsite în tabelele de perechi de funcții $[f(t), f^*(s)]$. Se poate recurge la calculul numeric uzând de o aproximare a funcției în domeniul timp, de pildă cea prin segmente de linie dreaptă. Transformarea Laplace inversă se poate face, de asemenea, numeric. Pentru aceasta se reaminteste aici formula lui Stehfest

$$f(t) + eroare = \frac{\ln 2}{t} \sum_{i=1}^N V_i f^* \left(i \frac{\ln 2}{t} \right)$$

în care N este un număr par arbitrar dar legat de precizia de calcul (numărul de digiti semnificativi cu care se operează) și în care valorile funcției transformate $f^*(s)$ sunt însumate ponderat cu ponderile

$$V_i = (-1)^{\frac{N}{2}+i} \sum_{k=\lceil \frac{i+1}{2} \rceil}^{\min(i, \frac{N}{2})} \frac{k^{\frac{N}{2}} (2k)!}{(\frac{N}{2} + k)! k! (k-1)! (i-k)! (2k-i)!}$$

pentru fiecare moment t luat în considerare. Metoda a fost propusă și aplicată în una din lucrările de la disciplina *Modelarea și simularea dinamicii sistemelor*.

A doua posibilitate de solutionare a problemei modelării constă în simularea Monte Carlo, adică în generarea de durate de viață aleatoare care să respecte legea de repartitie aleasă (Weibull) și în înlocuirea probabilităților diverse cu frecvențele relative ale observațiilor, frecvențe care tind în probabilitate către acele probabilități teoretice. Generarea se realizează pe calea cunoscută. Se utilizează un generator de numere (pseudo)aleatoare x repartizate uniform în intervalul (0,1) și se rezolvă repetat ecuația în t , $F(t) = x$, unde $F(t)$ este funcția de repartitie a duratei de viață, în cazul particular al repartitiei Weibull

$$F(t) = 1 - e^{-\frac{t^\beta}{\alpha}}$$

Se recomandă revederea pe scurt a *Lucrării 1* din acest *Ghid*.

O a treia posibilitate, poate cea mai comodă, face apel la funcții implementate în *Matlab*. Funcțiile **prefix + pdf** [+ **cdf** etc.] din subpașchetul **stats**, cu **prefix = weib** (de la Weibull) etc. și funcția **conv** facilitează calculele necesare în lucrarea prezentă. Funcțiile grafice pe care pașchetul *Matlab* le detine oferă posibilități de reprezentare în

imagini foarte instructive a unor functii legate de fiabilitatea sistemelor cu schimbare. Se recomandă utilizarea intensivă a acestor posibilități.

5. Chestiuni de studiu

Se cer a fi studiate si/sau evaluate următoarele:

- media duratelor între două schimbări succesive
- funcția de reînnoire $H(t)$ si/sau funcția densitate a reînnoirilor $h(t)$
- probabilitatea ca în decurs de 500 de unități de timp să se producă cel mult trei reînnoiri
- probabilitatea ca în decurs de 500 de unități de timp să se producă cel puțin trei reînnoiri
- probabilitatea ca în decurs de 500 de unități de timp să se producă exact trei reînnoiri

LUCRAREA 4

FIABILITATEA SISTEMELOR *SOFTWARE*

1. Obiectivele lucrării

- Studiul fiabilității programelor de calcul pe exemple simulate
- Estimarea unor parametri ai modelelor de fiabilitate a produselor *software*

2. Aparatură si suport documentar

- Calculatoare PC în configurație obișnuită
- Prezentul *Ghid de lucrări*
- Notele de curs de la disciplina *Fiabilitate si diagnoză* si orice alte surse bibliografice din domeniu
- Manuale de programare, documentarea *on-line (Help)*

3. Breviar teoretic

Între modelele fiabilității programelor de calcul modelele datorate lui Jelinski și Moranda sunt foarte populare printre specialiștii în studii de fiabilitate *software*.

Modelele Jelinski-Moranda au rata de defectare proporțională cu numărul de erori latente și este constantă în intervalul dintre două manifestări succesive ale erorilor din program.

Modelul clasic presupune că înaintea incidentului numărul k rata defectării are expresia

$$z(t) = \lambda_k = \varphi N(t) = \varphi (N - k + 1)$$

Se consideră că la fiecare oprire din cauza unei erori se și remediază acea eroare.

Modelul geometric utilizează o expresie diferită pentru rata defectării

$$z(t) = \begin{cases} \lambda_0 = \varphi N & t \in [0, t_1) \\ \lambda_1 = \varphi c N = c \lambda_0 & t \in [t_1, t_2) \\ \lambda_2 = \varphi c^2 N = c^2 \lambda_0 & t \in [t_2, t_3) \\ \dots & \dots \\ \lambda_k = \varphi c^k N = c^k \lambda_0 & t \in [t_k, t_{k+1}) \\ \dots & \dots \end{cases}$$

și la fiecare “pană” se rezolvă o fracție, $(1 - c)$, din erorile existente în program.

În ambele variante modelul contine câte doi parametri, φ si N în cazul clasic, c si λ_0 în cazul scăderii geometrice a numărului de erori.

Parametrii modelelor se pot estima din observatii experimentale, ceea ce poate crea premisa unor evaluări predictive asupra comportării unui produs *software* în timpul verificărilor sau în exploatare, fără a conduce testările până la ultima eroare din program.

Dacă observatiile experimentale sunt duratele de functionare între două întreruperi succesive datorate erorilor din program, x_1, x_2, \dots, x_n , atunci estimarea parametrilor N si φ , ca si a parametrilor c si λ_0 se poate face prin alegerea potrivită a unei functii de verosimilitate si maximizarea acesteia în raport cu parametrii de estimat.

Pentru modelul Jelinski-Moranda original, clasic, într-unul, oricare din intervalele dintre două defectări, densitatea de repartitie pentru x_k este

$$f_k(x_k / N, \varphi) = \varphi (N - k + 1) e^{-\varphi (N - k + 1) x_k}$$

Densitatea de repartitie pentru vectorul observatiilor x_1, x_2, \dots, x_n presupuse independente este

$$f(x_1, x_2, \dots, x_n / N, \varphi) = \varphi^n \prod_{k=1}^n (N - k + 1) e^{-\sum_{k=1}^n \varphi (N - k + 1) x_k}$$

si este tocmai functia de verosimilitate. Logaritmul acestei functii este de asemenea o functie de verosimilitate care este mai convenabilă pentru maximizat

$$L(x_1, x_2, \dots, x_n / N, \varphi) = n \ln \varphi + \sum_{k=1}^n \ln(N - k + 1) - \varphi \sum_{k=1}^n (N - k + 1)x_k$$

Anularea derivatelor parțiale ale acestei funcții conduce la un sistem de două ecuații neliniare în necunoscutele N și φ

$$\frac{\partial L}{\partial N} = \sum_{k=1}^n \frac{1}{N - k + 1} - \varphi \sum_{k=1}^n x_k = 0$$

$$\frac{\partial L}{\partial \varphi} = \frac{n}{\varphi} - \sum_{k=1}^n (N - k + 1)x_k = 0$$

Soluțiile rezultate \hat{N} și $\hat{\varphi}$ sunt estimările maximum verosimile ale parametrilor teoretici N și φ .

Analog, pentru modelul geometric, funcția densitate de probabilitate pentru una din variabilele x_k este

$$f_k(x_k / c, \lambda_0) = \lambda_0 c^{k-1} e^{-\lambda_0 c^{k-1} x_k}$$

funcția de verosimilitate are expresia

$$f(x_1, x_2, \dots, x_n / c, \lambda_0) = \lambda_0^n \left(\prod_{k=1}^n c^{k-1} \right) e^{-\lambda_0 \sum_{k=1}^n c^{k-1} x_k}$$

și logaritmul ei este

$$L(x_1, x_2, \dots, x_n / c, \lambda_0) = n \ln \lambda_0 + \ln c \left[\sum_{k=1}^n (k-1) \right] - \lambda_0 \sum_{k=1}^n c^{k-1} x_k$$

pentru care maximumul se localizează prin rezolvarea sistemului în c și λ_0

$$\frac{\partial L}{\partial c} = 0, \quad \frac{\partial L}{\partial \lambda_0} = 0$$

Literatura din domeniu sustine că, adesea, tendinta celor doi parametri de estimat este ca unul să devină foarte mare, celălalt să tindă către zero. În asemenea împrejurări este necesară prelungirea observatiilor asupra programului, asadar adăugarea de noi valori în secvența x_1, x_2, \dots, x_n .

4. Modul de lucru

Cu *script*-ul Matlab

```
clear  
  
rand('state',sum(100*clock))  
  
ner=100;  
  
fi=0.0001;  
  
for k=1:50  
    lam=fi*(ner-k+1);  
    u=rand;  
    x(k)=-log(1-u)/lam;  
  
end
```

se generează o secvență aleatoare de valori x_1, x_2, \dots, x_n asociată unui program care contine $N = 100$ de erori (**ner**), iar coeficientul φ din modelul Jelinski-Moranda este 0,0001 (**fi**). “Experimentul” este condus numai până la a 50-a întrerupere.

Pe baza unor porțiuni mai scurte din secvența generată, apoi din ce în ce mai lungi, se evaluează prin metoda verosimilității maxime parametrii modelelor. Se rezolvă, asadar, sistemele de ecuații rezultate prin anularea derivatelor parțiale.

Pentru modelul Jelinski-Moranda geometric se pot utiliza aceleași date “experimentale” sau se poate scrie o secvență Matlab generatoare asemănătoare celei de mai sus (procedură recomandată).

5. Chestiuni de studiu

- Se evaluează \hat{N} și $\hat{\Phi}$ și se compară cu valorile $N = 100$ și $\phi = 0,0001$
- Se evaluează $\hat{\lambda}_0$ și \hat{c} , se compară cu valori teoretice dacă acestea există (cazul *script*-ului Matlab specific)
- Se compară calitativ cele două modele
- Se observă tendința estimărilor pentru parametrii modelelor de a deveni extrem-de-mare – extrem-de-mic, tendință semnalată în literatură, și efectul pe care îl are îmbogățirea materialului experimental

LUCRAREA 5

ANALIZA COMPONENTELOR PRINCIPALE (ACP) ÎN DIAGNOZĂ

1. Obiectivele lucrării

- Studiul datelor experimentale corelate prin analiza componentelor principale (ACP)
- Realizarea algoritmică a aproximărilor prin direcții principale
- Studiul unor cazuri simulate sau reale legate de funcționarea unor instalații automatizate

2. Aparatură și suport documentar

- Calculatoare PC în configurație obișnuită
- Prezentul *Ghid de lucrări*

- Notele de curs de la disciplina *Fiabilitate si diagnoză* si orice alte surse bibliografice din domeniu
- Manuale de programare, documentarea *on-line (Help)*

3. Breviar teoretic

Analiza componentelor principale (ACP) este una din cele mai răspândite metode statistice de analiză a sistemelor descrise de varabile multiple. În esență, variabilele observate, afectate uzual de zgomote si mutual corelate statistic sunt reduse la o multime de variabile latente care reprezintă un gen de sumar al informatiei relevante din datele recoltate. Operatia se realizează prin proiectarea informatiei brute pe un subspatiu de dimensiune mai redusă. ACP este o procedură de a explica întreaga variatie a datelor observate, grupate într-o matrice $X \in R^{m \times n}$, cu m si n numărul de observatii, respectiv numărul de variabile observate. Descompunerea conformă ACP este dată de

$$X = t_1 p_1^T + t_2 p_2^T + \dots + t_k p_k^T + E$$

unde t_i si p_i sunt, pentru orice $i = 1, 2, \dots, k$, vectori score/loading, iar E este o matrice de diferente reziduale. Vectorii t_i sunt ortogonali si de lungime egală cu unitatea. Vectorii p_i la rândul lor sunt ortogonali, dar ei nu mai sunt si unitari. Prima componentă principală este aceea care explică cea mai mare parte a variatiilor: $t_1 = X p_1$. În spatiul n -dimensional

al vectorilor p_i , vectorul p_1 este pe directia celei mai mari variabilități și vectorul t_1 este alcătuit din proiecțiile vectorului de observații pe directia dată de p_1 .

A doua componentă principală este aceea care este a doua ca importantă/magnitudine. Ea este, ca și prima, o combinație liniară a vectorilor variabilelor observate și este ortogonală în raport cu prima. Mai departe, componentele sunt ordonate descrescător, în ordinea contribuției lor la variația generală a datelor observate. Pentru o matrice X de rang n se pot stabili asemenea componente. Dacă sunt corelații puternice și zgomot important atunci se retin $k < n$ astfel de componente, care sunt suficiente uzual pentru a explica variabilitatea datelor observate. Celelalte componente sunt la nivelul zgomotelor și prin eliminarea lor, procedură obișnuită, se obține un benefic efect de filtrare.

Calculul componentelor principale se poate face pe căi diverse. Una din posibilități constă în determinarea așa-numitelor valori singulare ale matricei X și apoi descompunerea acesteia conform relației

$$X = U\Sigma V^T = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \dots + \sigma_n u_n v_n^T$$

în care valorile singulare sunt ordonate crescător $\sigma_1 < \sigma_2 < \dots < \sigma_n$. Prima componentă principală este $\sigma_1 u_1$, primul vector “loading” este v_1 .

Algoritmul alternativ este cel al celor mai mici pătrate parțiale, este iterativ și neliniar. Acest algoritm calculează pe rând fiecare componentă

principală. Perechea t_i, p_i este calculată din matricea X , celelalte din matricile rezidualelor rezultate la fiecare etapă

$$X = t_1 p_1^T + E_1$$

$$E_1 = t_2 p_2^T + E_2$$

și analog în continuare. Algoritmul în detaliu conține pași de mai jos și se aplică pentru fiecare componentă principală înlocuind în pașii următori X cu $E_i, i = 1, 2, \dots, n-1$:

- (1) Se selectează un vector x_j din X și se redenumeste $t_1 : t_1 = x_j$
- (2) Se calculează $p_1 : p_1 = t_1^T X / t_1^T t_1$
- (3) Se normalizează $p_1 : p_1 = p_1 / \|p_1\|$
- (4) Se calculează $t_1 : t_1 = X p_1 / p_1^T p_1$
- (5) Se compară vectorul t_1 utilizat în pasul (2) cu vectorul t_1 calculat la pasul (4); dacă sunt sensibil aceiași ca valori calculul se încheie, dacă nu se reia cu pasul (2)

4. Modul de lucru

Se elaborează programul și se verifică pentru matrici de date X de dimensiuni reduse. Se propune, de pildă, stabilirea componentelor principale pentru o matrice de date în care liniile succesive conțin două elemente care sunt suma, respectiv diferența variabilelor u și v , prin urmare valori care sunt vădit dependente una de alta. Analog se pot

genera matrici de date/observatii X sub forma altor combinatii de variabile “de bază” similare celor notate mai devreme cu u si v . Se sugerează ca datele să fie generate aleator cu relatia binecunoscută $t_j = \sigma_j \sqrt{-2 \ln(r_1)} \sin(2\pi r_2)$, care asigură o lege de repartitie normală de medie nulă si de dispersie σ_j^2 dacă r_1 si r_2 sunt valori aleatoare repartizate uniform pe intervalul $(0, 1)$. Abaterile medii pătratice pot fi calculate în prealabil cu relatia $\sigma_j = 5r_3$, cu r_3 o altă serie de valori aleatoare provenite dintr-o distributie uniformă pe intervalul $(0, 1)$.

5. Chestiuni de studiu

- Eficienta algoritmului propus pentru stabilirea componentelor principale, sub aspectul vitezei de calcul
- Raportarea evaluărilor la cele asteptate, adică compararea numărului de directii principale cu cele stabilite la crearea/generarea matricilor de observatii X .

LUCRAREA 6

RETELELE NEURONALE ARTIFICIALE ÎN DIAGNOZĂ

1. Obiectivele lucrării

- Studiul topologiei unor rețele neuronale artificiale stratificate
- Instruirea rețelelor neuronale artificiale
- Studiul capacității rețelelor neuronale artificiale de a detecta și de a diagnostica disfuncționalitățile

2. Aparatură și suport documentar

- Calculatoare PC în configurație obișnuită
- Prezentul *Ghid de lucrări*
- Notele de curs de la disciplina *Fiabilitate și diagnostic* și orice alte surse bibliografice din domeniu

- Manuale de programare, documentarea *on-line* (*Help*)

3. Breviar teoretic

Retelele neuronale artificiale (RNA) sunt sisteme de celule de calcul similare neuronilor naturali, care împrumută de la mediul biologic topologii, funcții de răspuns la anumite intrări, funcții și praguri de activare, capacitatea de a învăța etc. Ca și celulele nervoase naturale, celulele unei RNA pot avea capacitatea de a se structura și de a executa calcule paralele de amploare. Structurile pot fi atât ciclice cât și stratificate. În lucrarea prezentă rețelele sunt de tipul stratificat, adică ele nu au reveniri: ieșirea unei celule, nu poate fi, direct sau mijlocit, intrare pentru o celulă care la rândul ei generează direct sau mijlocit o intrare pentru celula în discuție. La rețelele stratificate se disting un strat de celule de intrare, un strat de ieșire și unul sau mai multe straturi ascunse. Legăturile între straturi sunt numai într-un sens: dinspre intrare către ieșire, de la un strat la următorul, niciodată invers sau între celulele aceluiași strat.

Stratul de intrare are uzual un rol de centrare și scalare a intrărilor numerice care pot fi foarte diferite ca ordin de mărime și ca variație relativă. Așadar, neuronii artificiali din primul strat au funcții de activare/de răspuns de cele mai multe ori liniare.

Funcțiile de răspuns ale celorlalte celule componente ale unei RNA pot fi de diverse forme, în raport cu scopul urmărit. În această lucrare funcția intrare-iesire recomandată este cea sigmoidală

$$y_j = f(x_j) = \frac{1}{1 + e^{-\lambda x_j}}$$

în care indicele j marchează numărul neuronului, iar intrarea x_j este o intrare compusă conform relației

$$x_j = \sum_i w_{ji} y_i + b_j$$

prin ponderarea cu coeficienții w_{ji} a ieșirilor neuronilor din stratul precedent sau, după caz, a intrărilor rețelei. Adăosul b_j exprimă particularitatea celulei de a mentine o intrare fictivă sistematică (*bias*) chiar dacă toate intrările efective sunt nule, intrare care poate fi asociată uneori cu pragul de sensibilitate. Constanta λ din funcția intrare-iesire este un parametru care poate modifica panta funcției în limite largi, ceea ce permite aducerea funcției sigmoide oricât de aproape de funcția prag caracteristică neuronilor naturali dar rămânând încă o funcție derivabilă.

Derivabilitatea funcției sigmoide poate fi importantă pentru instruirea rețelelor neuronale artificiale stratificate. Instruirea în sine constă în stabilirea ponderilor w_{ji} și a pragurilor exprimate prin numerele b_j , pe baza unei multimi de învățare sau de instruire. Multimea de învățare este un set de perechi intrări-iesiri de proveniență experimentală. Prin aplicarea concomitentă sau succesivă a intrărilor, se modifică după o anumită

regulă ponderile până când este minimizat un criteriu de distanță între ieseșirile experimentale și ieseșirile calculate cu rețeaua. Prin urmare este vorba de minimizarea unei funcții, operație care poate utiliza metodele de gradient care presupun lucrul cu funcții derivabile.

În lucrarea prezentă se recomandă utilizarea metodei *propagării inverse* (*back propagation*) care combină un criteriu pătratic de instruire cu metoda gradientului. Mai în detaliu, criteriul este de forma

$$\sum_i (t_i - o_i)^T (t_i - o_i)$$

în care indicele i parcurge toate perechile intrări-ieseșiri experimentale, cu ieseșirile t_i experimentale și cu ieseșirile o_i obținute prin aplicarea acelușiei intrări rețelei instruite. Criteriul este, evident, o funcție de ponderile utilizate în rețea și de pragurile celulelor neuronale. Răspunsurile t_i , o_i sunt în general vectori, dar pot fi și variabile simple. În cazul vectorial, în produsul scalar din expresia de mai sus se poate intercala eventual o matrice Q de ponderi

$$\sum_i (t_i - o_i)^T Q (t_i - o_i)$$

Metodele de gradient amintite mai devreme orientează căutarea în spațiul ponderilor și al pragurilor pe direcția dată de derivatele parțiale ale criteriului propus. Este un proces iterativ și la fiecare iterație modificările efective ale ponderilor și pragurilor sunt combinații liniare ale diferentelor $(t_i - o_i)$, care sunt toate înmulțite cu un așa-numit factor de învățare α . Mai mult, calculul se realizează în etape, la început pentru

stratul neuronal de iesire, în continuare pentru stratul penultim s.a.m.d. către intrarea rețelei. La fiecare etapă diferențele de la iesirea unui strat se transformă în diferențe în iesirile stratului precedent. De aici calificativul de propagare inversă adoptat pentru metoda de instruire în discuție.

Singurele derivate care trebuie calculate sunt cele ale funcției sigmoidale. Acestea au o expresie simplă, $\lambda y(1 - y)$ și sunt funcții explicitate exclusiv de iesirea (generică) y a celulei.

Ponderile se pot “acorda” pe realitatea datelor de instruire și *aleator și adaptiv*. Asta înseamnă:

- Initializarea aleatoare a ponderilor și pragurilor și reținerea acestui punct din spațiul explorat ca centru al unui hiper-cub
- Evaluarea criteriului de distanță experiment-calcul și memorarea valorii lui
- Generarea la întâmplare a unui nou set de ponderi și praguri în hiper-cubul definit mai devreme și evaluarea criteriului de distanță
- Dacă noua valoare a criteriului de distanță este mai bună sau cel puțin la fel de bună ca precedentă, se memorează aceasta și punctul din spațiul ponderi-praguri, care este cel mai bun până la această etapă și care devine centrul hiper-cubului; dacă, dimpotrivă, valoarea este mai slabă nu se întreprinde nimic special; se reia procedura de la punctul anterior.

Căutarea se încheie când după un număr de evaluări nu se produce nici o nouă apropiere experiment-retea neuronală.

4. Modul de lucru

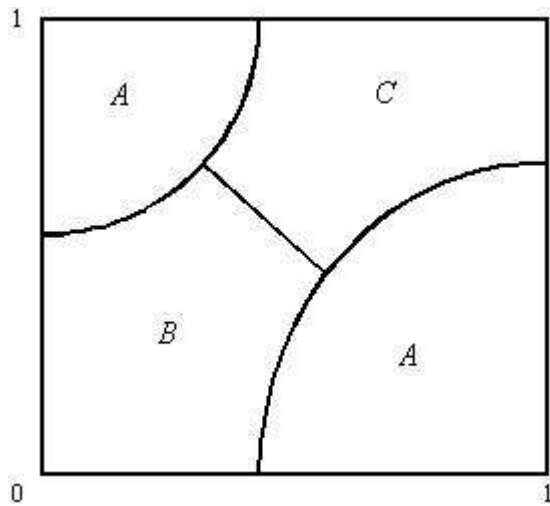
Pentru efectuarea lucrării sunt necesare datele experimentale care alcătuiesc multimea de învățare. Pentru aceasta se presupune că starea unui sistem este descrisă de două “simptome” cantitative, x_1 și x_2 , pozitive și subunitare. În planul (x_1, x_2) sunt delimitate trei diagnostice, A , B și C , întocmai ca în figura alăturată

În cuvinte, zonele-diagnostic sunt definite de conturul diagramei, de două sferturi de cerc de raze 0,3 și 0,6 cu centrele în vârfurile $(0, 1)$, respectiv $(1, 0)$ ale pătratului și de un segment din diagonala pătratului unitate.

Puncte generate aleator în pătratul unitate se vor situa în zonele A , B sau C . Acestea sunt etichetate corespunzător și asociate vectorilor $(1\ 0\ 0)^T$, $(0\ 1\ 0)^T$, respectiv $(0\ 0\ 1)^T$. Se generează în acest mod 50-100 de puncte care constituie multimea de învățare. Analog se poate genera mai târziu o multime, uzual mai restrânsă, pentru verificare.

Funcțiile de activare ale celulelor neuronale din straturile următoare celui de intrare sunt presupuse sigmoide cu parametrul λ egal cu unitatea.

Este permisă atribuirea pentru λ și a altor valori pozitive.



Se cere a se proceda la instruirea unei rețele neuronale cu două intrări și trei ieșiri, cu unul sau două straturi ascunse, cu câte 5-10 celule fiecare.

5. Chestiuni de studiu

- Algoritmul/algoritmii de instruire
- Viteza de învățare/instruire și convergența instruirii în general și în funcție de parametrul α dacă se folosește algoritmul propagării inverse
- Pe baza multimii de verificare generată după metoda enunțată mai sus se vor face aprecieri asupra eficacității diagnosticării cu ajutorul rețelei (rețelelor) neuronale instruite

