

---

```

clear;
%pack;
addpath(genpath('C:\FER\5TH SEMESTER\OBRADA_INF\Lab3\HMMall'))

% =====
% Oznacavanje stanja HMM modela
% Imamo tri pristrane kocke od kojih uvijek bacamo jednu odabranu
% Stanja modela su indeksi koristene pristrane kocke
% Vektor inicijalne vjerojatnosti stanja (za t=1)
% odredjen bacanjem nepristrane kocke:
prior0=[
1 % Prva kocka (ako je palo '1')
2 % Druga kocka (ako je palo '2' ili '3')
3 % Treca kocka (ako je palo '4', '5' ili '6')
]/6;
% Broj stanja HMM modela
Q=size(prior0,1);
% -----
% Matrica vjerojatnosti promjena stanja
%
% a11 a12 a13
% a21 a22 a23
% a31 a32 a33
% Za eksperiment sa stohastickom izmjenom stanja, parametar
% M se koristi za definiranje vjerojatnosi prijelaza u
% novo stanje u matrici prijelaza A, pri cemu se stanja nuzno
% mijenjaju ciklicki radi forsirane strukture tranzicijske matrice.
M= 9; % Ovdje definirate M iz vasesg personaliziranog zadatka.
% Formiraj matricu vjerojatnosti prijelaza stanja
% (uz ciklicku strukturu izmjene stanja, jer su
% prijelazi 1->3, 2->1 i 3->2 zabranjeni)

%Pod-zadatak 1 - Cjelovito definiranje HMM modela u Matlabu
%Temeljem zadanih ucestalosti pojedinih ishoda bacanja pristranih kocki i
temeljem
% zadanog parametra M u vasem Moodle zadatku, potrebno je dopuniti predlozak
Matlab
% skripte kako bi cjelovito opisali zadani HMM model ovog eksperimenta
ukljucujuci i
% matricu vjerojatnosti osmatranja izlaznih simbola.

transmat0=[
8 1 0 % P(1|1) P(2|1) P(3|1)
0 8 1 % P(1|2) P(2|2) P(3|2)
1 0 8 % P(1|3) P(2|3) P(3|3)
]/M;

% uoceni nizevi
O1 =[ 3 3 3 4 4 5 5 5 3 4 1 1 1 1 1 4 1 2 4 1 4 5 1 6 1 1 1 6 4 1 2 6 6 1 3
3 3 6 3 1 1];
O2 = [ 2 2 6 4 6 2 2 2 6 4 2 5 1 2 6 5 6 6 6 1 2 5 4 1 1 4 6 1 3 6 5 6 6 6 1
6 6 1 3 1 6];

```

---

---

```

B_count = [
20 3 1 6 4 6
4 4 20 2 3 7
2 5 5 6 20 2
];

% računanje sume svakog reda matrice B
row_sums = sum(B_count, 2);

% Podijeli svaki red sa sumom reda
obsmat0 = B_count ./ row_sums;

%Pod-zadatak 2 - Odredjivanje log-izvjesnosti osmatranja zadanog izlaznog
niza simbola za zadani model
% IZVJEŠTAJ :Da bi se objasnilo zašto je drugi niz manje izvjestan od prvog,
% trebalo bi pogledati vjerojatnosti ishoda u oba niza.
% Ako drugi niz sadrži više rijetkih ishoda (prema matrici vjerojatnosti
osmatranja),
% to bi mogao biti razlog za manju izvjesnost.

log_prob_01 = dhmm_logprob(O1, prior0, transmat0, obsmat0);
log_prob_02 = dhmm_logprob(O2, prior0, transmat0, obsmat0);

% Log šanse za niz 01 i za niz 02
fprintf('Log Šansa niza 01: %f\n', log_prob_01);
fprintf('Log Šansa niza 02: %f\n', log_prob_02);

% 2.b podzadatak
ratio = exp(log_prob_01 - log_prob_02);
fprintf('Drugi niz je ovoliko puta izgledniji od prvog niza %f\n', ratio);

obslik1 = multinomial_prob(O1,obsmat0);
obslik2 = multinomial_prob(O2,obsmat0);

% izračunaj unaprijednu alfu i unazadnu betu, gamma se ne koristi
[alpha, beta, gamma, ll] = fwdback(prior0, transmat0, obslik1, 'scaled', 0);

%Pod-zadatak 3 - Izracunavanje vjerojatnosti unaprijed i unazad za sva
skrivena stanja modela i sve
% vremenske trenutke osmatranja
%Za prvu sekvencu iz pod-zadatka 2 potrebno je primijeniti algoritme
"Unaprijed" i "Unazad" i izracunati
% unaprijedne vjerojatnosti  $\alpha_t(\text{stanje})$ 
% i unazadne vjerojatnosti  $\beta_t(\text{stanje})$ 
% za sve trenutke osmatranja  $t=1 \dots T$  za zadani model L.

%Vazno: pri pozivu funkcije ne smijete aktivirati skaliranje vjerojatnosti,
tj. u pozivu funkcije morate
% definirati ..., 'scaled', 0); kao sto je ucinjeno i u primjeru u uputama.
%Upisite koji iznos unaprijedne vjerojatnosti ste dobili za  $\alpha(2)$ 
% za  $t=27$  u prvo polje , odnosno iznos unazadne vjerojatnosti za  $\beta(1)$ 
% za  $t=8$  u drugo polje u eksponencijalnom zapisu.

```

---

---

```

%odredi  $\alpha_t(1)$  za  $t=27$ 
fprintf('αt(1): \n');
alpha(1,27)
% odredi  $\beta(2)$  za  $t=12$ 
fprintf(' β(2): \n');
beta(2,12)

% IZVJEŠTAJ: Alfa vjerojatnosti (unaprijed) mogu se koristiti za
izračunavanje ukupne
% vjerojatnosti osmatranja niza do određenog trenutka, dok beta
vjerojatnosti (unazad)
% pružaju ukupnu vjerojatnost osmatranja određenog trenutka do kraja niza.
Kombiniranjem
% ove dvije vjerojatnosti dobiva se ukupna vjerojatnost cijelog niza.

%Pod-zadatak 4 - Dekodiranje skrivenih stanja pomocu Viterbi algoritma

%Potrebno je primjenom Viterbi algoritma odrediti najizvjesniji niz
skrivenih stanja modela za prvi
% osmotreni niz iz drugog pod-zadatka. U narednih šest polja upisite
dekodirana stanja modela za prva
% tri i za zadnja tri vremenska koraka prve opservacije

%odredi najizvjesniji niz skrivenih stanja modela za prvi osmotreni niz iz
drugog pod-zadatka
vpath_o1 = viterbi_path(prior0,transmat0,obslik1);
vpath_o1 % procitaj prva tri i zadnja tri znaka

vpath_o2 = viterbi_path(prior0,transmat0,obslik2);

[l11, p1] = dhmm_logprob_path(prior0, transmat0, obslik1, vpath_o1);
cp1=cumprod(p1);

[l12, p2] = dhmm_logprob_path(prior0, transmat0, obslik2, vpath_o2);
cp2=cumprod(p2);

fprintf('LL1: %f\n', l11);
fprintf('LL2: %f\n', l12);

%Pod-zadatak 5 - Odredjivanje log-izvjesnosti osmatranja uzduz dekodiranih
Viterbi puteva
%razlika log-izvjesnosti preko svih puteva i log-izvjesnosti uzduz Viterbi
puta za oba osmotrena niza

%Ponovite odredjivanje Viterbi niza stanja i za drugi osmotreni niz iz pod-
zadatka 2, te za oba
% niza izracunajte log-izvjesnosti osmatranja ali samo uzduz dekodiranih ?
optimalnih? Viterbi puteva.
% Usporedite dobivene rezultate s onima iz pod-zadatka 2 gdje je izracunata
ukupna log-izvjesnost za
% sve moguće puteve skrivenih stanja. U naredna dva polja upisite razliku
log-izvjesnosti preko svih
% puteva i log-izvjesnosti uzduz Viterbi puta za oba osmotrena niza

```

---

---

```

fprintf('LL1 - logprob1: %f\n', -ll1+log_prob_01);
fprintf('LL2 - logprob2: %f\n', -ll2+log_prob_02);

%IZVJEŠTAJ:Rezultati dobiveni kroz alfa i beta vjerojatnosti trebali bi se
uskladiti s ukupnim
% log-izvjesnostima iz pod-zadatka 2, jer oba pristupa računaju istu krajnju
vjerojatnost niza.

%IZVJEŠTAJ:Predznak razlike između ukupne log-izvjesnosti i log-izvjesnosti
uzduž Viterbi puta
% može ukazivati na to koliko je Viterbi put reprezentativan za ukupne
vjerojatnosti niza. Ako
% je razlika negativna, to može značiti da Viterbi put nije najizvjesniji
put.Izračunavanje ž
% izvjesnosti osmatranja duž svih mogućih pojedinačnih puteva za cjelovite
nizove
% je teoretski moguće, ali u praksi je to često neizvodljivo zbog
eksponencijalnog
% broja puteva.

%Pod-zadatak 6 - Odredjivanje izvjesnosti osmatranja za skraceni niz i
najizvjesniji pojedinačni putevi stanja
%6a
%Za prvi osmotreni niz iz pod-zadatka 2 potrebno je odrediti ukupnu
izvjesnosti osmatranja skracenog niza,
% tj. samo za prva četiri osmotrena izlazna simbola o1, o2, o3 i o4. U tu
svrhu trebete iskoristiti ranije
%rjesenje iz trećeg pod-zadatka u kojem ste odredili sve vjerojatnosti
modela, ali za cjelovit niz.

%odredi ukupnu izvjesnosti osmatranja skracenog niza, tj. samo za prva
četiri osmotrena izlazna simbola
alpha(1,4) + alpha(2,4) + alpha(3,4);

%6b
%udio izvjesnosti osmatranja (normirano na 1) se ostvaruje uzduž Viterbi
puta u odnosu na sve moguće puteve stanja ovog modela

O1_short = [ 3 3 3 4];
obslik1_short = multinomial_prob(O1_short,obsmat0);
vpath_o1_short = viterbi_path(prior0,transmat0,obslik1_short);

%6c
% nadjeni Viterbi put stanja za prva četiri osmotrena simbola prvog niza
vpath_o1_short;

%6d
% izvjesnosti osmatranja prva četiri izlazna simbola, ali uzduž svih mogućih
pojedinačnih puteva resetke stanja

% Matrica svih mogućih puteva
num_states = 3; % broj stanja
num_obs = 4; % broj observacija
num_paths = num_states^num_obs; % ukupan broj puteva

```

---

---

```

% inicijaliziraj matricu za spremanje svih puteva
all_paths = zeros(num_paths, num_obs);

% generiraj sve puteve
tot = 0;
path_counter = 1;
for i = 1:num_states
    for j = 1:num_states
        for k = 1:num_states
            for l = 1:num_states
                all_paths(path_counter, :) = [i j k l];
                path_counter = path_counter + 1;
                tot = tot + 1;
            end
        end
    end
end

% inicijaliziraj listzu šansi
llm = zeros(num_paths, 1);

% izračunaj log vjerojatnosti za svaki put
for i = 1:num_paths
    [llm(i), p] = dhmm_logprob_path(prior0, transmat0, obslik1_short,
all_paths(i,:));
end

fprintf("Broj stanja: %f\n",tot);

%6e
% koliko puteva od svih njih uopce nisu moguci
num_inf = sum(llm == -Inf);
fprintf('Broj nemogućih puteva: %d\n', num_inf);

%6f
%koji udio ukupne izvjesnosti osmatranja (normirano na 1) se kumulativno
ostvaruje uzduz prvih pet
% najizvjesnijih puteva ove sortirane liste

[sorted_llm, sorted_indices] = sort(llm, 'descend');
probabilities = exp(sorted_llm);
total_probability = sum(probabilities);
top_5_probability = sum(probabilities(1:5));
cumulative_proportion = top_5_probability / total_probability;
fprintf('Proporcija top 5 puteva od svih mogućih: %f\n',
cumulative_proportion);

%IZVJEŠTAJ :alpha(1,4) + alpha(2,4) + alpha(3,4), odnosno unaprijedne
%vjerojatnosti svih obzervacija za dani model lambda.
%IZVJEŠTAJ :Viterbi rješenje za skraćeni niz ne smije se koristiti iz
prethodnih
% pod-zadataka jer se ono odnosi na drugačiji skup podataka (kratki niz
umjesto cijelog niza).

```

---

---

```

%IZVJEŠTAJ :Među putevima stanja, neki mogu imati istu izvjesnost, ovisno o
vjerojatnostima
% tranzicija i osmatranja. Skraćeni Viterbi put je jedan od mogućih puteva,
a njegova izvjesnost
% ovisi o specifičnom skupu tranzicija i osmatranja
%IZVJEŠTAJ :

%Pod-zadatak 7 - Generiranje opservacija za zadani model

%Generirajte višestruke slučajne nizove osmotrenih izlaznih simbola s
%nex=14
% razlicitih nizova, pri čemu svaki niz treba biti duljine T=135 vremenskih
uzoraka.
% Za generiranje podataka koristiti funkciju dhmm_sample u skladu s uputama,
uz parametre HMM modela iz
% vases individualnog pod-zadatka 1. Sacuvajte ovu matricu opservacija jer
ce biti intenzivno koristen a i
% u narednim pod-zadacima. Prije poziva funkcije, svakako resetirajte
generator slučajnih brojeva na
% pocetnu vrijednost naredbom rng('default').

nex = 14; %broj nizova
T = 125; %broj vremenskih uzoraka
rng('default')
data = dhmm_sample(prior0, transmat0, obsmat0, nex, T);

%Pod-zadatak 8 - Odredjivanje dugotrajne statistike osmotrenih simbola i
usporedba s njihovim teorijskim ocekivanjima

%Za nizove koji su generirani u pod-zadatku 7, potrebno je eksperimentalno
odrediti vjerojatnosti
% osmatranja svih izlaznih simbola koristenjem slicnih primjera iz uputa. Za
prvu osmotrenu sekvencu
% iz proslog pod-zadatka upisite broj osmatranja svakog izlaznog simbola, od
1 do 6

%IZVJEŠTAJ:Dugotrajne vjerojatnosti pojedinih stanja i izlaznih simbola
ukazuju na stabilnu
% distribuciju u koju HMM konvergira s vremenom. Degenerirani HMM model s
jednakim dugotrajnim
% statistikama osmatranja izlaznih simbola imao bi iste vjerojatnosti za
svako stanje i svaki
% izlazni simbol.

% IZVJEŠTAJ:Empirijske dugotrajne vjerojatnosti osmatranja simbola dobivene
su usrednjavanjem
% broja pojava preko svih eksperimenata, a približno se podudaraju s
teorijskim dugotrajnim
% vjerojatnostima kako bi se procijenila preciznost modela.

% 8 a
%prva osmotrena sekvenca
hm = hist(data', [1,2,3,4,5,6]);

```

---

---

```

hm % gledaj prvi stupac za prvu sekvencu

% 8b
% teorijska očekivanja dugotrajnih vjerojatnosti osmatranja izlaznih simbola
pi_stac=transmat0;
for i=1:125
    pi_stac=pi_stac*transmat0;
end

stationary_dist = pi_stac(1, :);
stationary_obs_prob = zeros(1, 6);

for symbol = 1:6
    stationary_obs_prob(symbol) = sum(stationary_dist .* obsmat0(:,
symbol)');
end

fprintf('Dugotrajna vjerojatnost stanja 1: %f\n', stationary_dist(1));
fprintf('Dugotrajna vjerojatnost uočavanja stanja 4: %f\n',
stationary_obs_prob(1));

stationary_obs_prob;

% 8c
%empirijske dugotrajne vjerojatnosti osmatranja simbola
symbol_counts = hist(data', 1:6);
empirical_probs = sum(symbol_counts, 2) / (nex * T);

abs_diff = abs(empirical_probs - stationary_obs_prob');
max_dif = max(abs_diff);
%ajveci apsolutni iznos razlike izmedju empirijskih i teorijskih vjerojatnost
% izlaznih simbola maksimiziran preko svih 6 izlaznih simbola
fprintf('Maximalna apsolutna razlika u stanjima: %f\n',max_dif );

%Pod-zadatak 9 - Izracun log-izvjesnosti osmatranja pojedinacnih generiranih
opservacija temeljem zadanog modela
% Izracunaj u petlji log-izvjesnosti svakog niza

%Za svaki od slucajnih nizova koji su generirani u pod-zadatku 7 potrebno je
izracunati log-izvjesnost
% osmatranja uz zadani model, tj. uz isti model koji je koristen za
generiranje ovih osmatranja. Nakon
% toga izracunajte najveću, najmanju i srednju vrijednost log-izvjesnost
usrednjenu preko svih nex osmotrenih
% nizova, te upisite dobivene rezultate u naredna tri polja (max, min i mean)

%IZVJEŠTAJ:Razlike u izvjesnostima pojedinih nizova mogu proizaći iz
različitih
% distribucija osmatranja i tranzicija unutar tih nizova.

nex=14; % Broj eksperimenata
llm=zeros(nex,1); % Stupac log-izvjesnosti

```

---

---

```

for i=1:nex
    llm(i)=dhmm_logprob(data(i,:), prior0, transmat0, obsmat0);
end

max_llm = max(llm);
min_llm = min(llm);
mean_llm = mean(llm);

fprintf('Max: %f\n',max_llm);
fprintf('Min: %f\n',min_llm );
fprintf('Mean: %f\n',mean_llm );

%Pod-zadatak 10 - Provedite postupak treniranja parametara HMM modela

%Temeljem svih nizova osmatranja koji su generirani u pod-zadatku 7,
potrebno je izracunati dva nova
% HMM modela primjenom funkcije dhmm_em. Vazno: u oba slucaja ogranicite
broj iteracija EM postupka na
% najvise 200, a prag relativne promjene izvjesnosti u odnosu na proslu
iteraciju za zavrsetak postupka
% postavite na 1E-6.

%Za prvi HMM model inicijalizacija parametara modela za pocetnu iteraciju EM
postupka treba biti potpuno
% slucajna (prema uputama), uz prethodno resetiranje generatora pseudo-
slucajnih brojeva na pocetnu
% vrijednost. Za drugi HMM model za inicijalizaciju EM postupka iskoristite
parametre zadanog modela.
% Tocnost vaseg izracuna parametara modela verificirat ce se u narednom pod-
zadatku.

%IZVJEŠTAJ:Razlika u broju iteracija potrebnih za treniranje dva HMM modela
može ukazivati na
% razlike u početnim uvjetima i konvergenciji modela tijekom treniranja.

% 10a
%određivanje koji broj iteracija je bio potreban za estimaciju parametara
HMM modela EM postupkom za oba modela
rng('default')
num_states = size(transmat0, 1);
num_obs_sym = size(obsmat0, 2);

random_prior = normalize(rand(num_states, 1));
random_transmat = normalize(rand(num_states, num_states), 2);
random_obsmat = normalize(rand(num_states, num_obs_sym), 2);

[LL1, prior1, transmat1, obsmat1] = dhmm_em(data, random_prior,
random_transmat, random_obsmat, 'max_iter', 200, 'thresh', 1E-6);

[LL2, prior2, transmat2, obsmat2] = dhmm_em(data, prior0, transmat0,
obsmat0, 'max_iter', 200, 'thresh', 1E-6);

%Pod-zadatak 11 - Usporedna evaluacija zadanog modela, slucajnog modela i

```

---



---

```

treniranih modela na istim podacima koji su korišteni za trening
% Potrebno je usporediti uspješnost modeliranja opservacijskih nizova
generiranih u pod-zadatku 7
% sa svim raspoloživim HMM modelima, izračunom log-izvjesnosti osmatranja
svih generiranih nizova
% funkcijom dhmm_logprob. Kao "los" model za usporedbu, potrebno je
koristiti HMM model s potpuno
% slučajnim parametrima, koji je korišten za inicijalizaciju prvog od dva
nova "optimalna" HMM modela
% u prošlom pod-zadatku (Vazno:, ... pazite da su parametri ovog slučajnog
modela uistinu generirani
% odmah nakon inicijalizacije generatora pseudo-slučajnih brojeva).

%IZVJEŠTAJ:Log-izvjesnosti pojedinih nizova odražavaju koliko dobro svaki
niz odgovara modelu.
% Uspoređivanjem rezultata novih modela s log-izvjesnostima istih nizova za
zadani model, može
% se procijeniti koliko su novi modeli uspješni u opisivanju podataka.

%IZVJEŠTAJ:Provjera estimiranog modela na istim podacima koji su korišteni
za treniranje nije
% uvijek primjeren postupak jer može doći do prenaučenosti (overfittinga).
Pravi postupak
% treniranja i validacije uključuje odvajanje skupa podataka na trening i
test skupove.

nex = size(data, 1);

log_likelihood_given = zeros(nex, 1);
log_likelihood_random = zeros(nex, 1);
log_likelihood_first_new = zeros(nex, 1);
log_likelihood_second_new = zeros(nex, 1);

for i = 1:nex
    log_likelihood_given(i) = dhmm_logprob(data(i,:), prior0, transmat0,
obsmat0);
    log_likelihood_random(i) = dhmm_logprob(data(i,:), random_prior,
random_transmat, random_obsmat);
    log_likelihood_first_new(i) = dhmm_logprob(data(i,:), prior1, transmat1,
obsmat1);
    log_likelihood_second_new(i) = dhmm_logprob(data(i,:), prior2,
transmat2, obsmat2);
end

% Summarize the results
sum_ll_given = sum(log_likelihood_given);
sum_ll_random = sum(log_likelihood_random);
sum_ll_first_new = sum(log_likelihood_first_new);
sum_ll_second_new = sum(log_likelihood_second_new);

% Display the results
fprintf('Log-likelihood for the given model: %f\n', sum_ll_given);
fprintf('Log-likelihood for the random model: %f\n', sum_ll_random);

```

---

---

```
fprintf('Log-likelihood for the first new model: %f\n', sum_ll_first_new);
fprintf('Log-likelihood for the second new model: %f\n', sum_ll_second_new);
```

```
Log Šansa niza 01: -62.118050
```

```
Log Šansa niza 02: -80.430924
```

```
Drugi niz je ovoliko puta izgledniji od prvog niza 89780106.618498
```

```
at(1):
```

```
ans =
```

```
1.419991191008368e-17
```

```
 $\beta(2):$ 
```

```
ans =
```

```
2.489377385383477e-22
```

```
vpath_o1 =
```

```
Columns 1 through 13
```

```
1      2      2      2      3      3      3      3      3      3      1      1      1
```

```
Columns 14 through 26
```

```
1      1      1      1      1      1      1      1      1      1      1      1      1
```

```
Columns 27 through 39
```

```
2      1      1      1      1      1      1      1      1      2      2      2      2
```

```
Columns 40 through 41
```

```
2      2
```

```
LL1: -64.769664
```

```
LL2: -85.006049
```

```
LL1 - logprob1: 2.651614
```

```
LL2 - logprob2: 4.575125
```

```
Broj stanja: 81.000000
```

```
Broj nemogućih puteva: 57
```

```
Proporcija top 5 puteva od svih mogućih: 0.980125
```

```
hm =
```

```
Columns 1 through 13
```

```
40    15     19     25     34     37     33     24     25     28     25     20     28
```

---

|    |    |    |    |    |    |    |    |    |    |    |    |    |
|----|----|----|----|----|----|----|----|----|----|----|----|----|
|    | 15 | 13 | 11 | 12 | 13 | 8  | 16 | 13 | 11 | 7  | 13 | 10 |
| 10 | 36 | 26 | 40 | 27 | 17 | 23 | 29 | 26 | 25 | 20 | 21 | 23 |
| 19 | 13 | 15 | 12 | 7  | 21 | 13 | 17 | 11 | 16 | 14 | 18 | 16 |
| 20 | 33 | 31 | 22 | 22 | 25 | 28 | 24 | 25 | 29 | 38 | 40 | 35 |
| 16 | 13 | 21 | 15 | 23 | 12 | 20 | 15 | 25 | 16 | 21 | 13 | 13 |
| 20 |    |    |    |    |    |    |    |    |    |    |    |    |

Column 14

35  
11  
24  
15  
20  
20

Dugotrajna vjerojatnost stanja 1: 0.333333  
Dugotrajnja vjerojatnost uočavanja stanja 4: 0.216667  
Maximalna apsolutna razlika u stanjima: 0.016143  
Max: -195.180987  
Min: -217.788944  
Mean: -204.615475  
iteration 1, loglik = -3079.062495  
iteration 2, loglik = -3052.791427  
iteration 3, loglik = -3052.120512  
iteration 4, loglik = -3051.466384  
iteration 5, loglik = -3050.796590  
iteration 6, loglik = -3050.075341  
iteration 7, loglik = -3049.258553  
iteration 8, loglik = -3048.287405  
iteration 9, loglik = -3047.079154  
iteration 10, loglik = -3045.513604  
iteration 11, loglik = -3043.413757  
iteration 12, loglik = -3040.521124  
iteration 13, loglik = -3036.474619  
iteration 14, loglik = -3030.827073  
iteration 15, loglik = -3023.185623  
iteration 16, loglik = -3013.591375  
iteration 17, loglik = -3003.007523  
iteration 18, loglik = -2993.154620  
iteration 19, loglik = -2985.272827  
iteration 20, loglik = -2979.245455  
iteration 21, loglik = -2974.101489  
iteration 22, loglik = -2968.785441  
iteration 23, loglik = -2962.360785  
iteration 24, loglik = -2953.881380  
iteration 25, loglik = -2942.398047  
iteration 26, loglik = -2927.638800  
iteration 27, loglik = -2911.317733  
iteration 28, loglik = -2896.802376

---

```
iteration 29, loglik = -2886.167841
iteration 30, loglik = -2878.965127
iteration 31, loglik = -2873.855642
iteration 32, loglik = -2869.900248
iteration 33, loglik = -2866.661653
iteration 34, loglik = -2863.958877
iteration 35, loglik = -2861.705427
iteration 36, loglik = -2859.845297
iteration 37, loglik = -2858.331536
iteration 38, loglik = -2857.118968
iteration 39, loglik = -2856.162184
iteration 40, loglik = -2855.416439
iteration 41, loglik = -2854.839936
iteration 42, loglik = -2854.395996
iteration 43, loglik = -2854.054196
iteration 44, loglik = -2853.790385
iteration 45, loglik = -2853.585946
iteration 46, loglik = -2853.426768
iteration 47, loglik = -2853.302233
iteration 48, loglik = -2853.204346
iteration 49, loglik = -2853.127072
iteration 50, loglik = -2853.065822
iteration 51, loglik = -2853.017090
iteration 52, loglik = -2852.978176
iteration 53, loglik = -2852.946992
iteration 54, loglik = -2852.921914
iteration 55, loglik = -2852.901673
iteration 56, loglik = -2852.885276
iteration 57, loglik = -2852.871939
iteration 58, loglik = -2852.861046
iteration 59, loglik = -2852.852110
iteration 60, loglik = -2852.844743
iteration 61, loglik = -2852.838639
iteration 62, loglik = -2852.833554
iteration 63, loglik = -2852.829293
iteration 64, loglik = -2852.825701
iteration 65, loglik = -2852.822654
iteration 66, loglik = -2852.820051
iteration 1, loglik = -2864.616653
iteration 2, loglik = -2855.751947
iteration 3, loglik = -2854.309082
iteration 4, loglik = -2853.785839
iteration 5, loglik = -2853.536700
iteration 6, loglik = -2853.404788
iteration 7, loglik = -2853.332054
iteration 8, loglik = -2853.291204
iteration 9, loglik = -2853.268028
iteration 10, loglik = -2853.254794
iteration 11, loglik = -2853.247203
iteration 12, loglik = -2853.242835
iteration 13, loglik = -2853.240315
Log-likelihood for the given model: -2864.616653
Log-likelihood for the random model: -3079.062495
```

---

*Log-likelihood for the first new model: -2852.817814*  
*Log-likelihood for the second new model: -2853.238858*

*Published with MATLAB® R2023b*