FINAL EXAM

AAEC 6564
INSTRUCTOR: KLAUS MOELTNER

General Instructions

Please type everything in LaTeX (including all Math) and hand in a pdf file. For the Matlab part, answer questions in LaTeX, and attach your script, log file, and any graphs to your main pdf file.

IMPORTANT: *Independent work* is required for this assignments - no group collaboration, please.

Question 1: Approximating a truncated normal via Accept-Reject Sampling

Consider the normal density with **mean 1** and **variance 1**, truncated to the $[0, 4]$ region. I will refer to it henceforth as the *target density*.

PART A

(1) Using the formulas given in the module 14 lecture notes, compute the expectation, variance, and standard deviation of the truncated distribution (feel free to use Matlab to do the actual computing). Please be precise to the fourth decimal.

(2) Compute the integrating factor c_f for this target density, and show the form of its kernel (call it $\tilde{f}(\theta)$).

(3) You would like to use acceptance sampling to approximate this target distribution. Consider the $U[0, 4]$ uniform as a source density. Show its integrating constant c_s and its kernel $\tilde{s}(\theta)$.

(4) Compute the importance multiplier $\tilde{M} = \underset{0 \leq \theta \leq 4}{max} \frac{\tilde{f}(\theta)}{\tilde{s}(\theta)}$ and the analytical acceptance rate.

(5) Using script *mod14_AR2* as an example, write a Matlab script for an acceptance sampler with 10,000 accepted draws. Make sure to set your random seeds for *rand* and *randn* to "37" at the beginning of your script (as usual).

Capture the mean and standard deviation of the kept draws and write them into an Excel table, along with the true (analytical) values for the target density. Also add your results for the analytical and empirical acceptance rates.

PART B

(1) Stay in the *same script*, do *NOT re-set the random seed*.

(2) Now consider a new source density, that is an **UN**-truncated normal with (mean = mean of truncated target), and (variance = **2 *** variance of truncated target). Compute the integrating constant for this source and show its kernel.

(3) Compute the importance multiplier and the analytical acceptance rate.

(4) Implement an acceptance sampler with 10,000 accepted draws.
(Hint: Make sure to discard all draws from the source that fall outside the [0,4] interval.)
Capture the mean and standard deviation of the kept draws and add them into your Excel table, along with the analytical and empirical acceptance rates.

(5) Plot the true target density and the two approximations in the same graph. Comment on the relative performance of the approximations. Use the following code to obtain the draws from the true density for plotting:

```
pd=makedist('Normal',1,1);
t=truncate(pd,0,4);
ygrid = linspace(0,4,R)';
ytrue = pdf(t,ygrid);
```

Question 2: Approximating a truncated normal via a 3-Component Mixture Model

Now suppose you want to approximate the same target density as before with a mixture-of-normals distribution, using a Gibbs Sampler.

(1) Start a *new script*, and *re-set the seeds to 37*. Take 10,000 draws from the truncated normal, using the following code:

```
n=10000; %sample size
pd=makedist('Normal',1,1);
t=truncate(pd,0,4);
y=random(t,n,1);
```

(2) Using the *mod10_2CMMMix* script as an example, write a Gibbs /sampler for a THREE-component mixture-of-normals model. You will have to adjust the Gibbs Sampler function *gs_2cmm* to accommodate the third component (call the modified function *gs_3cmm*). Use the following starting values, priors, and tuners:

```
% general elements
r1=10000; % burn-ins
r2=10000; % keepers
R=r1+r2;
% generic OLS
bols=inv(X'*X)*X'*y;
res=y-X*bols;
s2=(res'*res)/(n-k);
% prior for betas
mu0=0;
V0=eye(k)*10;
betadraw=bols;
% prior for sig2's
sig2draw=s2;
v0=1/2;
tau0=1/2;
%prior for cp
acp0=[2;2;2];
%draw starting vector of cp
cpdraw=[1/3;1/3;1/3];
%draw starting vector of indicators
Zdraw=mnldraws(repmat(cpdraw',n,1));
%n by 3 - my own function to draw from the MNL - much faster than Matlab's
```

(3) In the *same script*, and *without re-setting the random number seed*, generate mixed prediction vector *yp* as in script *mod10_2CMMplots* and capture its mean and standard deviation. Add these values to your Excel table from the previous question.

(4) Plot *yp* against the true distribution (generate the true draws as you did in the previous question). Comment on the plot and compare it to that from the previous question. How does the 3CMM compare to the A-R approaches from Q1 in fitting the target density?

Question 3: Approximating a truncated normal via Metropolis-Hastings Sampling

Of course, you can also use MH to approximate a free-standing density (not just conditional posteriors, as we have done so far). All you need is the kernel of the otherwise unknown target density, plus truncation bounds, if any. Here we will consider a simple RWC version of the MH to approximate the truncated normal

(1) Start a *new script*, and *re-set the seeds to 37*. Take again 10,000 draws from the truncated normal, using the following code:

```
n=10000; %sample size
pd=makedist('Normal',1,1);
t=truncate(pd,0,4);
y=random(t,n,1);
```

(2) Loop over R=100,000 iterations, with starting draw taken at random from the target density. Set the tuner std for the MH to 1:

```
R=100000;
ydraw=random(t,1,1);
ystd=1;
```

(3) At each iteration, draw a candidate *ycan* from the simple normal with mean=*yold*, and std = *ystd*.

(4) After drawing *ycan*, add an "if" condition to tell Matlab to only proceed if *ycan* is within the truncation bounds of the target ([0,4]);

(5) Derive the logged acceptance probability and decide if you keep the new draw or stick with the old (note: (i) there are no priors involved here, just the kernel of the target density), (ii) if *ycan* falls outside the truncation bounds, you should automatically stick with *yold*.)

(6) While collecting your draws of y, also keep track of your acceptance count (= how often you accept the candidate).

(7) Capture the mean and std of your draws of y, as well as the acceptance rate (= acceptance count/ R) and add them to your Excel table from the previous question(s). Which approach provided the best approximation to the true moments? Please make sure to attach the Excel table to your output.

(8) Plot your drawn y's against the true distribution (generate the true draws as you did in the previous question). Comment on the plot and compare it to that from the previous questions. How does the MH approach compare to the previous ones in approximating the true target density?