Modeling Movie Life Cycles and Demand Model

Ali Bakhtiari

Introduction & Model Specification:

In this project we are trying to model the box-office sales of the movies using a data set of movie sales for a period of 190 weeks. The modeling part is based the paper by Ainslie et al. (2005)[1] in which they develop a combination of a *sliding-window logit model* and a gamma diffusion pattern in hierarchical Bayes framework. They use random effects logit with changing consideration sets combined with a gamma diffusion pattern to model the expected market share of each movie in each week and use the diffusion pattern to capture the market attractiveness of a particular movie in a particular week. Then using the parameter of gamma distribution they define some new parameters that represent the different aspects of market attractiveness. We will define them later in this project. The market attractiveness is also used as a predictor of box-office sales which is captured in a *hierarchical regression model*. They use Bayesian approach to estimate these two models and verify their estimates by predicting an out-of-sample model. In this project, we try to fit a hierarchical regression model to our data and verify our results with an out-of-sample prediction. In the following paragraph we briefly go through the model specification and define different parameters used in this model.

To model the market attractiveness we use a gamma diffusion pattern which can be parameterized as

$$N_i \frac{1}{\beta_i^{\alpha_i}\Gamma(\alpha_i)} t^{\alpha_i - 1} e^{t/\beta_i}$$

Where indices *i* and *t* represent a specific movie and week, respectively, and N_i is the total demand for the movie. We re-parameterize this as follows

[1] Ainslie, A., Dreze, X. and Zufryden, F., 2005, "Modeling Movie Life Cycles and Market Share", *Marketing Science*, 24(3), 508-517

$$\eta_i t^{\gamma_i / \beta_i} e^{(1-t)/\beta_i}$$

Where

$$\eta_i = N_i \frac{1}{\beta_i^{\alpha_i} \Gamma(\alpha_i) e^{1/\beta_i}}$$ and

$$\gamma_i = (\alpha_i - 1)\beta_i$$

We then use regression model to specify the weekly box-office sales (S_{it}) as a function of gamma diffusion pattern plus some error terms. The hierarchical regression model is defined as follows

$$S_{it} = \eta_i w_{it}^{\gamma_i / \beta_i} e^{(1-w_{it})/\beta_i} + \varepsilon_{it}$$

Where w_{it} represents the number of weeks in release of movie *i* in time period *t*, η_i is the expected attractiveness of the movie in its opening week, γ_i indicates when peak attractiveness occurs and β_i is a speed parameter representing how fast attractiveness builds or decays.

To link the three movie-level parameters (η_i, γ_i and β_i) to movie characteristics we set up a hierarchical regression model on the three parameters using a mix of continuous variables (Z_i). This linkage characterized as

$$\eta_i^* = \ln(\eta_i)$$

$$\gamma_i^* = \gamma_i$$

$$\beta_i^* = \ln(\beta_i)$$

$$\begin{bmatrix} \eta_i^* \\ \gamma_i^* \\ \beta_i^* \end{bmatrix} = Z_i\Delta + \Sigma_\Delta$$

$$\varepsilon_{it} \sim i.i.dN(0, \sigma_d^2)$$

Some notes about the dataset used for estimation

The dataset includes 190 weeks of box-office sales for 1843 movies. We use these movie characteristics as explanatory variables in the regression model: number of weeks on screen, rating (R,PG-13, ...), Release span, number of opening week screens, budget, IMDB rating, TV guide rating and EBERT rating. The movies who were on screen before week one are removed from the dataset. We also removed movies with less than 5 million dollars budget, less than 10 weeks on screens and those movies with less than 1000 screens. To handle scaling problem we use log of box-office sales in the model. The final dataset used for estimation has includes 405 movies.

Bayesian Estimation

The model is estimated using MCMC approach. In this section we describe the likelihood function of the model, state prior assumptions and form the joint posterior distribution. Then based on the posterior distribution we derive the conditional distributions of the parameters need to be estimated.

Based on the box-office sales regression model and the hierarchical regression model for the movie parameters we define the likelihood function as

$$f(S_{it} \mid \eta_i, \gamma_i, \beta_i) = \prod_i \left(\prod_t \left(\frac{1}{\sigma_d^2} \exp\left(- \left(S_{it} - \hat{S}_{it}\right)^2 \Big/ 2\sigma_d^2 \right)^{I_{it}} \frac{1}{\left|V_d^{-1}\right|^{i/2}} \exp\left(-\frac{1}{2} \left(\begin{bmatrix} \eta_i^* \\ \gamma_i^* \\ \beta_i^* \end{bmatrix} - Z_i \Delta \right)' V_d^{-1} \left(\begin{bmatrix} \eta_i^* \\ \gamma_i^* \\ \beta_i^* \end{bmatrix} - Z_i \Delta \right) \right) \right) \right)$$

where

$\hat{S}_{it} = \eta_i w_{it}^{\gamma_i / \beta_i} e^{(1-w_{it})/\beta_i}$ which is considered as the mean box-office sales distribution.

And

$$\begin{bmatrix} \hat{\eta}_i^* \\ \hat{\gamma}_i^* \\ \hat{\beta}_i^* \end{bmatrix} = Z_i \Delta$$

We assume non-conjugate distributions for parameters η_i, γ_i and β_i and thus they need to be estimated through Metropolis-Hasting method. But, for the hierarchical regression model we assume conjugate priors as follows and use Gibbs sampling:

1. Prior for σ_d^2 is inverse gamma with parameters $\alpha_0/2$ and $\nu_0/2$

   $p(\sigma_d^2) \propto (\sigma_d^2)^{-(\alpha_0/2+1)} e^{-\frac{\nu_0}{2\sigma_d^2}}$ where α_0 is 3 and ν_0 set to 1.

2. Prior for Δ is normal with mean 0 and variance A_d

$$p(\Delta) \propto e^{-\frac{1}{2}\Delta' A_d^{-1}\Delta}$$

Where $A_d = I(8)\times 200$

3. Prior for V_d^{-1} is inverse Wishart with parameters $V_0 = I(3)\times 0.01$ and $df = 7$

$$p(V_d^{-1}) \propto \left|V_d^{-1}\right|^{\frac{df-3-1}{2}} e^{-tr(V_0^{-1}V_d^{-1})/2}$$

Therefore the joint posterior distribution is

$$p(\eta_i,\gamma_i,\beta_i,\sigma_d^2,V_d^{-1},\Delta) \propto \prod_i \left(\prod_t \left(\frac{1}{\sigma_d^2}\exp\left(-\left(S_{it}-\hat{S}_{it}\right)^2 \Big/ 2\sigma_d^2 \right)^{I_{it}} \frac{1}{\left|V_d^{-1}\right|^{1/2}} \exp\left(-\frac{1}{2}\left(\begin{bmatrix} \eta_i^* \\ \gamma_i^* \\ \beta_i^* \end{bmatrix} - Z_i\Delta \right)' V_d^{-1} \left(\begin{bmatrix} \eta_i^* \\ \gamma_i^* \\ \beta_i^* \end{bmatrix} - Z_i\Delta \right) \right) \right) \right) \times$$

$$(\sigma_d^2)^{-(\alpha_0/2+1)} e^{-\frac{v_0}{2\sigma_d^2}} \times e^{-\frac{1}{2}\Delta' A_d^{-1}\Delta} \times \left|V_d^{-1}\right|^{\frac{df-3-1}{2}} e^{-tr(V_0^{-1}V_d^{-1})/2}$$

Now, based on the joint posterior distribution we form conditional posterior of parameters to be used in Gibbs sampling: (hereafter we call the vector of (η_i , γ_i , β_i) as $\underline{\theta}$.

1. The conditional posterior distribution for σ_d^2

$$p(\sigma_d^2 \,|\, \underline{\theta}, V_d^{-1}, \Delta) \propto \frac{1}{(\sigma_d^2)^{n/2}} \exp\left(-\sum_i \sum_t \left(S_{it} - \hat{S}_{it}\right)^2 \Big/ 2\sigma_d^2 \right) \times (\sigma_d^2)^{-(\alpha_0/2+1)} e^{-\frac{\nu_0}{2\sigma_d^2}}$$

$$\propto (\sigma_d^2)^{\left(\frac{\alpha_0+n}{2}\right)} e^{-\left(\frac{\nu_0+\sum_i\sum_t\left(S_{it}-\hat{S}_{it}\right)^2}{2\sigma_d^2}\right)}$$

$$\Rightarrow \sigma_d^2 \sim IG\left(\frac{\alpha_0+n}{2}, \frac{\nu_0+\sum_i\sum_t\left(S_{it}-\hat{S}_{it}\right)^2}{2}\right)$$

2. The conditional posterior distribution for Δ

$$p(\Delta \,|\, \underline{\theta}, V_d^{-1}, \sigma_d^2) \propto \exp\left(-\frac{1}{2} \left(\begin{bmatrix} \eta_i^* \\ \gamma_i^* \\ \beta_i^* \end{bmatrix} - Z_i\Delta \right)' V_d^{-1} \left(\begin{bmatrix} \eta_i^* \\ \gamma_i^* \\ \beta_i^* \end{bmatrix} - Z_i\Delta \right) \right) \times e^{-\frac{1}{2}\Delta' A_d^{-1}\Delta}$$

$$\propto e^{-\frac{1}{2}(\theta - Z\Delta)' V_d^{-1}(\theta - Z\Delta)} \times e^{-\frac{1}{2}\Delta' A_d^{-1}\Delta} \propto e^{-\frac{1}{2}(\theta' - \Delta' Z') V_d^{-1}(\theta - Z\Delta)} e^{-\frac{1}{2}\Delta' A_d^{-1}\Delta} \propto e^{-\frac{1}{2}\left(\theta' V_d^{-1}\theta - \theta' V_d^{-1} Z\Delta - \Delta' Z'\theta' V_d^{-1} + \Delta' Z' V_d^{-1} Z\Delta + \Delta' A_d^{-1}\Delta\right)}$$

$$\propto e^{-\frac{1}{2}\left(\Delta'(Z'V_d^{-1}Z + A_d^{-1})\Delta - \theta' V_d^{-1} Z\Delta - \Delta' Z' V_d^{-1}\theta' + \theta' V_d^{-1}\theta\right)}$$

$$\Rightarrow p(\Delta \,|\, \underline{\theta}, V_d^{-1}, \sigma_d^2) \sim N((Z'V_d^{-1}Z + A_d^{-1})^{-1} Z'V_d^{-1}\theta, (Z'V_d^{-1}Z + A_d^{-1})^{-1})$$

3. And finally the conditional posterior distribution for V_d^{-1}

$$p(V_d^{-1} \mid \underline{\theta}, \Delta, \sigma_d^2) \propto \frac{1}{\left|V_d^{-1}\right|^{n/2}} \exp\left(-\frac{1}{2} \sum_i \left(\begin{bmatrix} \eta_i^* \\ \gamma_i^* \\ \beta_i^* \end{bmatrix} - Z_i \Delta \right)' V_d^{-1} \left(\begin{bmatrix} \eta_i^* \\ \gamma_i^* \\ \beta_i^* \end{bmatrix} - Z_i \Delta \right) \right) \times \left|V_d^{-1}\right|^{-\frac{df}{2}} e^{-tr(V_0^{-1} V_d^{-1})/2}$$

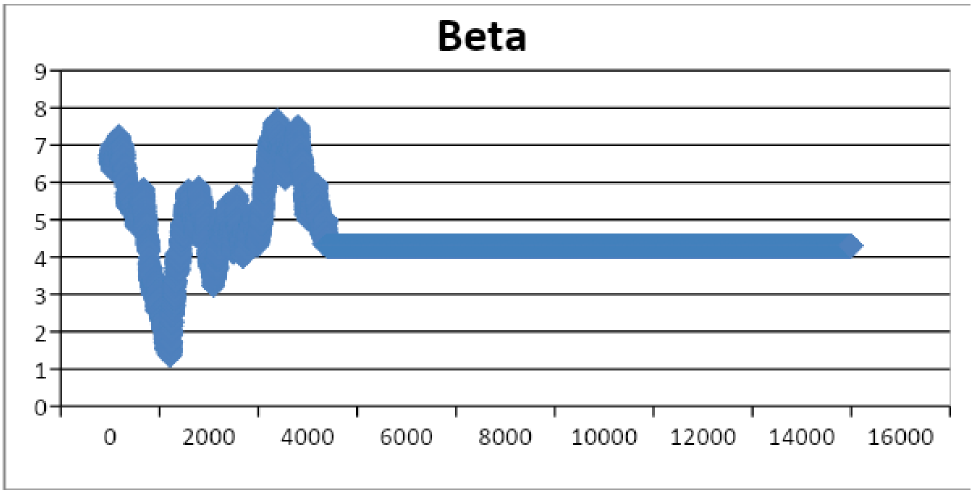
$$\propto \left|V_d^{-1}\right|^{\left(\frac{n+df}{2}\right)} e^{-tr(V_0^{-1} V_d^{-1})/2} \exp\left(-\frac{1}{2} tr\left(\sum_i \left(\begin{bmatrix} \eta_i^* \\ \gamma_i^* \\ \beta_i^* \end{bmatrix} - Z_i \Delta \right) \left(\begin{bmatrix} \eta_i^* \\ \gamma_i^* \\ \beta_i^* \end{bmatrix} - Z_i \Delta \right)' V_d^{-1} \right) \right)$$

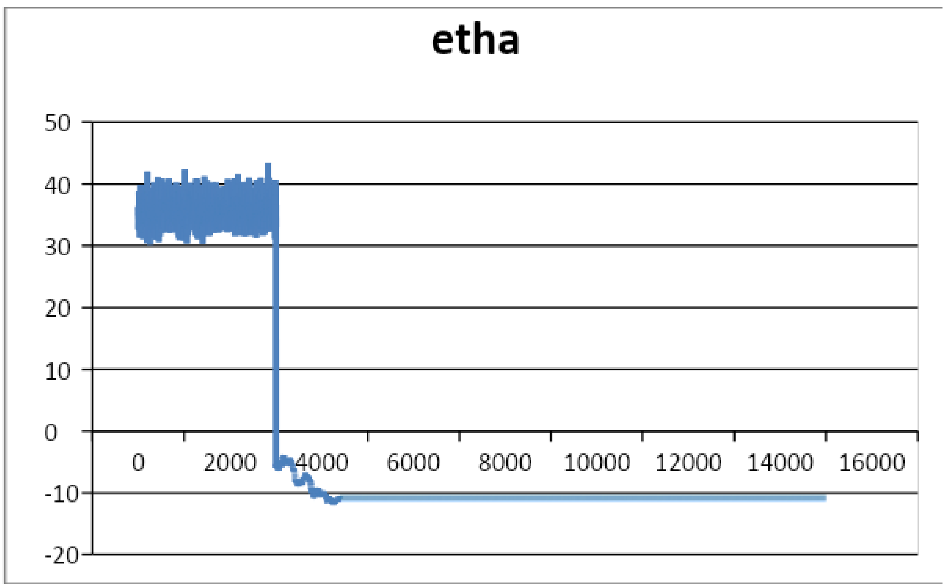
$$\propto \left|V_d^{-1}\right|^{\left(\frac{n+df}{2}\right)} e^{-\frac{1}{2} tr\left(V_0^{-1} + \sum_i \left(\begin{bmatrix} \eta_i^* \\ \gamma_i^* \\ \beta_i^* \end{bmatrix} - Z_i \Delta \right) \left(\begin{bmatrix} \eta_i^* \\ \gamma_i^* \\ \beta_i^* \end{bmatrix} - Z_i \Delta \right)' \right)}$$

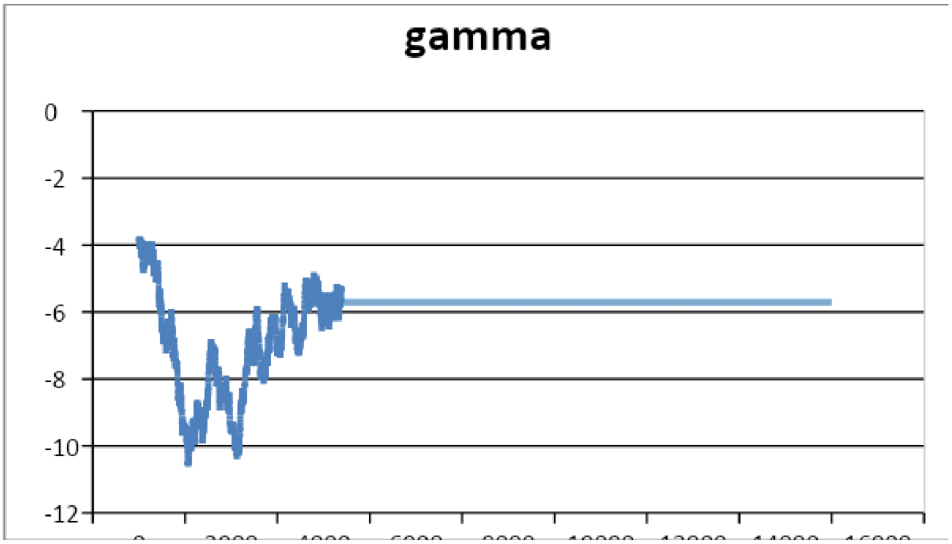
$$\Rightarrow p(V_d^{-1} \mid \underline{\theta}, \Delta, \sigma_d^2) \sim IW\left(n + df, V_0^{-1} + \sum_i \left(\begin{bmatrix} \eta_i^* \\ \gamma_i^* \\ \beta_i^* \end{bmatrix} - Z_i \Delta \right) \left(\begin{bmatrix} \eta_i^* \\ \gamma_i^* \\ \beta_i^* \end{bmatrix} - Z_i \Delta \right)' \right)$$

Bayesian Estimation and Analysis of Results

As mentioned before we use random-walk Metropolis-Hasting method for updating movie-level parameters (η_i, γ_i and β_i). The total number of draws is 20,000 out of which 5000 used as burn-in and 15,000 are stored. Totally, 8676 of draws are accepted which gives the acceptance rate of 43% that seems to be a reasonable. Below we have provided the graph of posterior of these parameters for one movie and the rest have the same shape. Obviously, the draws of all the three parameters converges after almost 4000 draws.


Beta
9
8
7
6
5
4
3
2
1
0
0
2000
4000
6000
8000
10000
12000
14000
16000


etha
50
40
30
20
10
0
-10
-20
0
2000
4000
6000
8000
10000
12000
14000
16000



For the parameters of the hierarchical regression model we use Gibbs sampling method based on the conditional posterior distribution of those parameters. The total number of draws is 4000 out which 1000 used as burn-in and 3000 draws stored for estimation. Below are the results of estimation and the graph of posterior draws:

$$V_d^{-1} = \begin{bmatrix} 1.9826 & 0.4064 & 0.1660 \\ 0.4064 & 0.8546 & 0.3458 \\ 0.1660 & 0.3458 & 0.5512 \end{bmatrix}$$

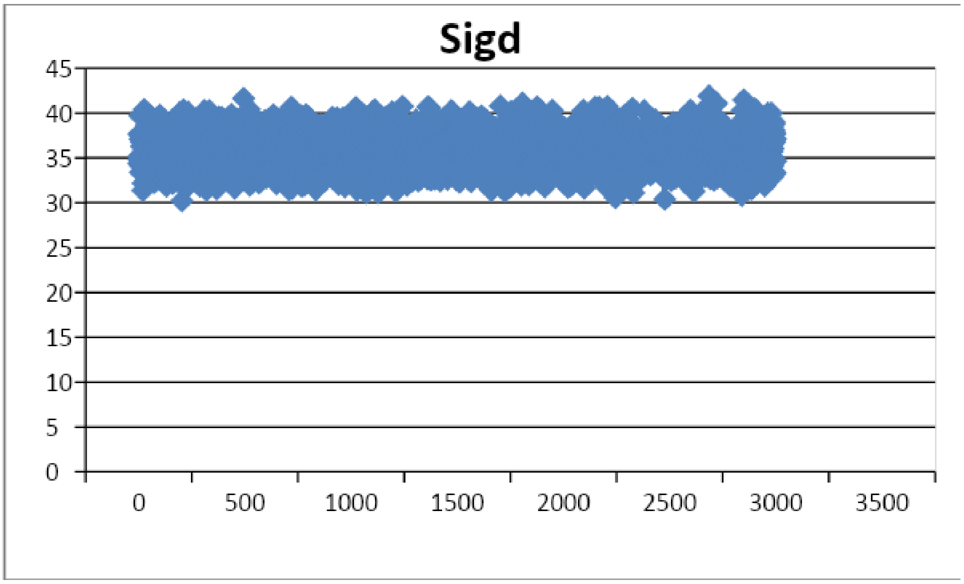
$$\sigma_d^2 = 35.7277$$

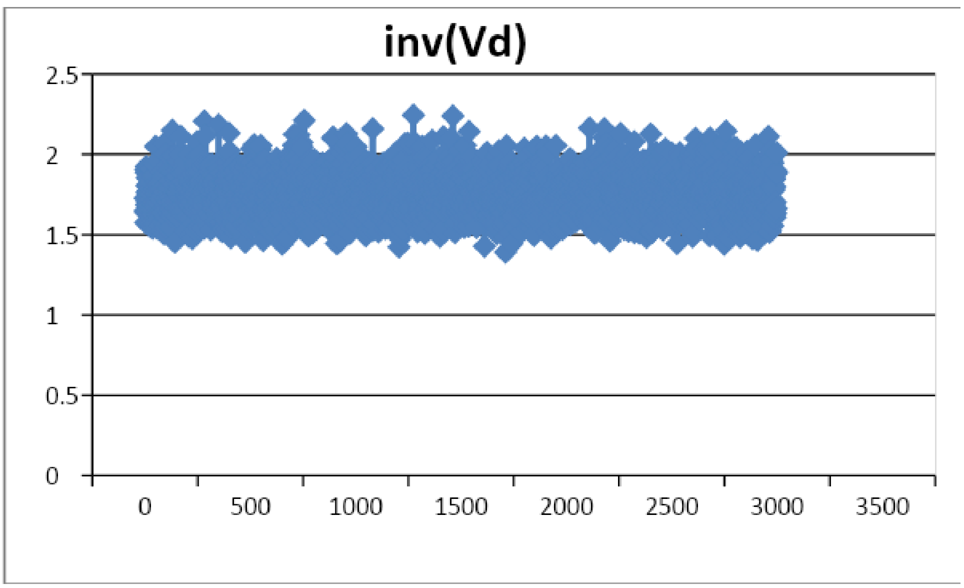
Type		η			γ			β	
Δ	Lower 95	Est.	Upper 95	Lower 95	Est.	Upper 95	Lower 95	Est.	Upper 95
Week	-0.2322	**-0.2124**	-0.1928	-0.1955	**-0.175**	-0.1552	-0.0029	0.0166	0.0368
Rating	0.4113	**0.4755**	0.5402	-0.1588	**-0.093**	-0.031	-0.1676	**-0.1052**	-0.0386
Release	-0.9634	**-0.8956**	-0.8265	-0.1204	-0.0511	0.0176	0.4545	**0.5231**	0.5925

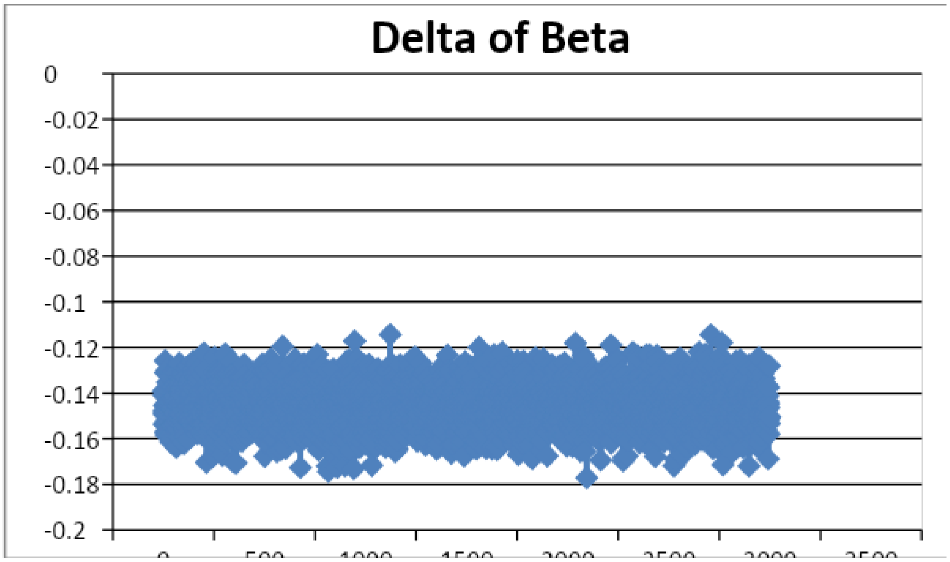
Screens	0.4099	**0.4333**	0.4558	0.0648	**0.0892**	0.1128	-0.0113	0.0129	0.0371
Budget	-0.0605	**-0.0513**	-0.0419	-0.0023	0.0071	0.0161	0.1291	**0.1386**	0.1477
IMDB rat.	-0.697	**-0.6328**	-0.5701	-0.2118	**-0.1472**	-0.0832	-0.5268	**-0.4634**	-0.3981
TV guide	0.5262	**0.5679**	0.6116	0.3718	**0.4154**	0.4593	0.1866	**0.2323**	0.2767
EBERT	0.8996	**0.9172**	0.9337	-0.3592	**-0.3421**	-0.3251	-0.1629	**-0.1455**	-0.1276

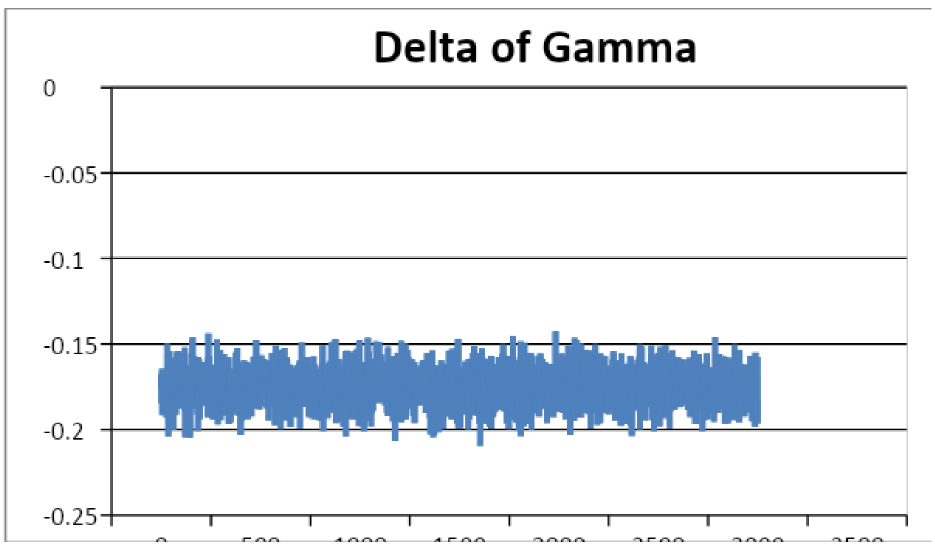
The above estimates shows that almost most of the variables are significant (they are bolded) at 95 percentile. These results shows that number of screens on opening day, the reviews that the movie gets on TV guide and EBERT and the rating of the movie have positive effects on expected attractiveness of the movie in its opening week. Rating and the reviews of the movie on the other hand reduces the time when the peak of attractiveness occurs. Widening the span of release accelerates the attractiveness of the movie and so does the amount of budget spent on production of the movie. In any case, all the movie characteristics have impacts (positive or negative) on the expected attractiveness of the movie on its opening week which shows the importance of first week of opening that is affected by more factors than the others.

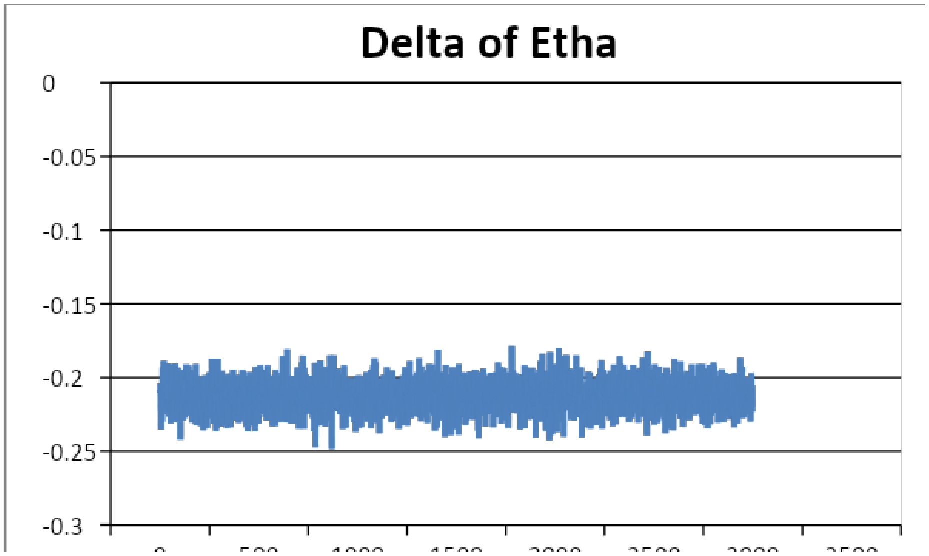
Below I have provided the posterior draws graph for each of some of these parameters (the rest have similar shapes). As we can see all of them show good convergence,


Sigd
45
40
35
30
25
20
15
10
5
0
0
500
1000
1500
2000
2500
3000
3500

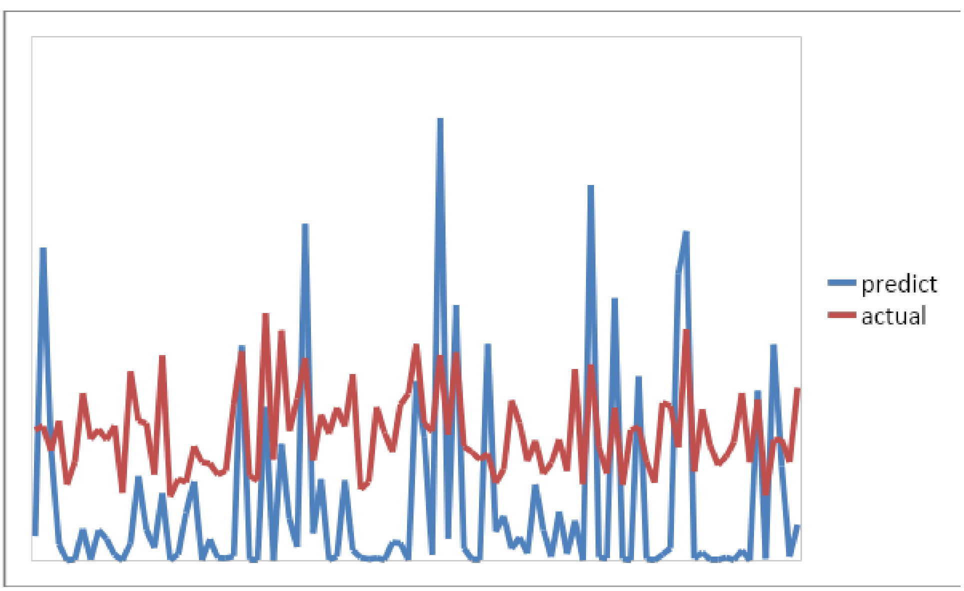

inv(Vd)
2.5
2
1.5
1
0.5
0
0
500
1000
1500
2000
2500
3000
3500


Delta of Beta
0
-0.02
-0.04
-0.06
-0.08
-0.1
-0.12
-0.14
-0.16
-0.18
-0.2


Delta of Gamma
0
-0.05
-0.1
-0.15
-0.2
-0.25



To check the performance of our model, we also performed an out-of sample analysis. We dropped the last 200 movies and estimated the model again with 205 movies. Then, using the estimates from sample we tried to predict the sales of the last 200 movies. The graph below shows the sum of sales for the out-of-sample compared with real sales data,



As we can see the prediction follows the same pattern but is a little bit off the actual values. It might be due to less number of observations. I guess that if we did prediction for the time dimension we would get better prediction.

Appendix

1. MATLAB code for making the dataset from raw data

```
%%%%%%%% scaling the movie sales %%%%%%%%

i = 1;
t = 1;

for i=1:1843
    for t=1:190
        if sraw(i,t) ~= 0
            sraw(i,t) = log(sraw(i,t));
        elseif sraw(i,t) == 0
            sraw(i,t) = 0;
        end
    end
    sraw(i,194)=sraw(i,194)/100;
end

s = sraw;

%%%% deleting movies started before week zero %%%%%%%%%

[a4 b4] =find(s(:,1) ~= 0);
s(a4,:)=[];
```

```
%%%%%%%% deleting movies which have less than 5 weeks on screens %%%%%%%%

[row col] = size(s);
j = 1;
t = 1;
count = 0;

while j <= row
    count = 0;
    for t=1:190
        if s(j,t) > 0
            count = count+1;
        end
    end
    if count <= 5
        s(j,:)=[];
    else
         j = j+1;
    end
    [row col] = size(s);

end

%%%%%% deleting movies with less than 1000 screens %%%%%%%%%%

[row3 col3] = size(s);
[a1 b1] =find(s(:,194)<10);
s(a1,:)=[];

%%%%%%% deleting movies with budget less than 5 million %%%%%%%

[a6 b6] = find(s(:,195)<5);
s(a6,:)=[];

%%%%%%% deleting rating = -1 %%%%%%%%

[row4 col4] = size(s);
[a2 b2] =find(s(:,194) == -1);
s(a2,:)=[];
```

2. MATLAB code for calculating the likelihood function

```
function[loglike] = loglikelihood(param1,param0,data,invV)
m = 205;
ethasi = zeros(m,1);
gamasi = zeros(m,1);
betasi = zeros(m,1);

%%%%% calculating shat %%%%%

ethai = zeros(m,1);
gamai = zeros(m,1);
betai = zeros(m,1);


ethasi = param1(:,1);
```

```
gamasi = param1(:,2);
betasi = param1(:,3);

ethai = exp(ethasi);
gamai = gamasi;
betai = exp(betasi);

j = 1;
t = 1;
shat = zeros(m,190);
ss = data(:,1:190);

for j=1:m
    w = 0;

    for t=1:190
        if ss(j,t) ~= 0
            w = w+1;

shat(j,t)=ethai(j,1)*w^(gamai(j,1)/betai(j,1))*exp((1-w)/betai(j,1));
        elseif ss(j,t) == 0
            shat(j,t) = 0;
        end
    end
end

i = 1;
t = 1;

for i=1:m
    for t=1:190
        if shat(i,t) > 1
            shat(i,t) = log(shat(i,t));
        elseif shat(i,t) <= 1
            shat(i,t) = 0;
        end
    end
end

%%%%%% calculating likelihood %%%%%

k = 1;
ll =1;

while k <= m
    ltwo = 1;
    ltwo =
exp(-1/2*((param1(k,:)'-param0(k,:)')'*invV*(param1(k,:)'-param0(k,:)')));

    lone = 1;

    while t <= 190
        if ss(j,t) ~= 0
            ind = 1;
            lone = (exp(-(ss(i,t)-shat(i,t))^2/(2*sigd^2)))^ind;
        elseif ss(j,t) == 0
            ind = 0;
```

```
            lone = 1;
        end
        t = t+1;
    end

    k = k+1;
    ll1 =(lone*ltwo);
    ll = ll*ll1;
    loglike = log(ll);

end
```

3. MATLAB code for Bayesian Estimation

```
%%%%% priors on delta %%%%%%%

m = 405;
ad = eye(8)*200;
tau = eye(3)*0.01;
df = 7;
deltai = zeros(8,3)+chol(inv(ad))*randn(8,3);
invV = iwishrnd(tau,df)*(df-3-1);
sigd = 10;


%%%%%% calculating thetahat %%%%%%%

theta1 = zeros(3,m);
thetahat = z*deltai;
thetahatmean = thetahat;
p = 1;
bi = 5000;
store = 15000;
n = store + bi;
cn = 0;
etham = zeros(store,m);
gamam = zeros(store,m);
betam = zeros(store,m);

while p <= n
    %%%%%% generate thetas from thetahat from a multivariate normal %%%%%
    i = 1;
    for i = 1:m
        thetahatp = thetahat';
        theta1(:,i) = thetahatp(:,i)+chol(invV)*randn(3,1);
    end

    %%%%%%% runnig MH to get updates for etha, gamma and beta %%%%%%%%

    thetas = theta1';

    loga =
-loglikelihood(thetas,thetahatmean,s,invV)+loglikelihood(thetahat,thetahatmean,
s,invV);
```

```
    if log(unifrnd(0,1)) <= loga
        thetahat = thetas;
        cn = cn + 1;
    else
        thetas = thetahat;
    end

    thetass = thetas';

    if p > bi
        etham(p-bi,:)=thetass(1,:);
        gamam(p-bi,:)=thetass(2,:);
        betam(p-bi,:)=thetass(3,:);
    end
    p = p+1;
end

%%%%%%%%%% Running Gibbs Sampling for sigd, Vd and Deltai %%%%%%%%%%%%%%%%

bi2 = 1000;
store2 = 3000;
n2 = store2 + bi2;
pp = 1;

sigdm = zeros(store2,1);
invVm1 = zeros(store2,3);
invVm2 = zeros(store2,3);
invVm3 = zeros(store2,3);
deltam1 = zeros(store2,8);
deltam2 = zeros(store2,8);
deltam3 = zeros(store2,8);

betami = mean(betam);
ethami = mean(etham);
gamami = mean(gamam);

betamup = betami';
gamamup = gamami';
ethamup = ethami';


%%%%%%% calculating shat updated %%%%%%%%

ethaup = exp(ethamup);
gamaup = gamamup;
betaup = exp(betamup);

j = 1;
t = 1;
shatup = zeros(m,190);
ss = s(:,1:190);

for j=1:m
    w = 0;

    for t=1:190
        if ss(j,t) ~= 0
```

```
            w = w+1;

shatup(j,t)=ethaup(j,1)*w^(gamaup(j,1)/betaup(j,1))*exp((1-w)/betaup(j,1));
        elseif ss(j,t) == 0
            shatup(j,t) = 0;
        end
    end
end

i = 1;
t = 1;

for i=1:m
    for t=1:190
        if shatup(i,t) > 1
            shatup(i,t) = log(shatup(i,t));
        elseif shatup(i,t) <= 1
            shatup(i,t) = 0;
        end
    end
end

%%%%%%%%%%%%% calculating s-squared %%%%%%%%%%%%%

i = 1;
t = 1;
ff = 0;
sq = 0;

for i = 1:m
       for t = 1:190
        if i == 20 || 85 %strcmp(shatup(i,t), 'NaN') ~= 1
              f = (ss(i,t))^2;

        else %i == 20 || 85 %strcmp(shatup(i,t), 'NaN') == 1
              f = (ss(i,t)-shatup(i,t))^2;
        end
        ff = ff + f;
       end
    sq = sq + ff;
end

alpha0 = 3;
nu0 = 1;

sigup = zeros(store2,1);
pp = 1;
thetaup = [ethamup gamamup betamup];

X = zeros(3*m,8);

ii = 1;
j = 1;

while j <= 3*m
    X(j,:)=z(ii,:);
    X(j+1,:)=z(ii,:);
```

```
    X(j+2,:)=z(ii,:);
    j = j+3;
    ii = ii+1;
end

Y = thetaup';

Y1 =[Y(1,:);Y(1,:);Y(1,:)];
Y2 =[Y(2,:);Y(2,:);Y(2,:)];
Y3 =[Y(3,:);Y(3,:);Y(3,:)];

%%%%%%%%%%% updating parameters %%%%%%%%%%%

while pp <= n2

    %%%%%%%%%%% updating sigd-squared %%%%%%%%%%%

    alpha = (alpha0+m);
    nu = sqrt((alpha0*nu0+m*sq)/(alpha0+m));
    sigdup = 1/(randg(alpha,1,1)/nu);

    %%%%%%%%%% updating invV %%%%%%%%%%%%%
    dfup = df + m;

    i = 1;
    tau2 = zeros(3,3);
    for i=1:m
    tau3 = (thetaup(i,:)-thetahat(i,:))'*(thetaup(i,:)-thetahat(i,:));
    tau2 = tau2 + tau3;
    end

    tauup = invV+tau2;
    invVup = iwishrnd(tauup,dfup);

    %%%%%%%%%% updating deltas %%%%%%%%%%%

    ee = 1;
    j = 1;
    mm11 = 0;
    nn11 = 0;
    mm22 = 0;
    nn22 = 0;
    mm33 = 0;
    nn33 = 0;

    while j <= 3*m
        mm1 = X(j:j+2,:)'*invVup*X(j:j+2,:);
        nn1 = X(j:j+2,:)'*invVup*Y1(:,ee);

        mm11 = mm11 + mm1;
        nn11 = nn11 + nn1;

        mm2 = X(j:j+2,:)'*invVup*X(j:j+2,:);
        nn2 = X(j:j+2,:)'*invVup*Y2(:,ee);

        mm22 = mm22 + mm2;
        nn22 = nn22 + nn2;
```

```
        mm3 = X(j:j+2,:)'*invVup*X(j:j+2,:);
        nn3 = X(j:j+2,:)'*invVup*Y3(:,ee);

        mm33 = mm33 + mm3;
        nn33 = nn33 + nn3;

        j = j+3;
        ee = ee+1;
    end

    deltamean1 = inv(mm11+inv(ad))*(nn11);
    deltaup1 = deltamean1 + chol(inv(mm11+inv(ad)))*randn(8,1);

    deltamean2 = inv(mm22+inv(ad))*(nn22);
    deltaup2 = deltamean2 + chol(inv(mm22+inv(ad)))*randn(8,1);

    deltamean3 = inv(mm33+inv(ad))*(nn33);
    deltaup3 = deltamean3 + chol(inv(mm33+inv(ad)))*randn(8,1);


    if pp > bi2

        sigdm(pp-bi2,1)=sigdup;
        invVm1(pp-bi2,:)=invVup(1,:);
        invVm2(pp-bi2,:)=invVup(2,:);
        invVm3(pp-bi2,:)=invVup(3,:);
        deltam1(pp-bi2,:)=deltaup1';
        deltam2(pp-bi2,:)=deltaup2';
        deltam3(pp-bi2,:)=deltaup3';

    end
    pp = pp+1;
end


'Vd Variance of the error terms in Hierarchical Regression'

invVm = [mean(invVm1);mean(invVm2);mean(invVm3)]

'Sigma: standard deviation of error term in Demand Model'

mean(sigdm)

'Coefficients for etha (expected attractiveness'

mean(deltam1)'

'Coefficients for gamma (peak of attractiveness)'

mean(deltam2)'

'Coefficients for beta (rate of attractiveness)'

mean(deltam3)'

%  95 percentile
```

```
qq11 = sort(deltam1);

betas11 = mean(deltam1);

m9511 = [qq11(ceil(store2*0.025),:);qq11(floor(store2*0.975),:)];

betapost11 = [qq11(ceil(store2*0.025),:);betas11;qq11(floor(store2*0.975),:)];

fprintf('\n');

disp('Posterior MCMC Estimates of etha Coeffs with 95% interval');

fprintf('\n');

fprintf('lower\t\t     mean\t\t     upper\n');

fprintf('\n');

fprintf('%9.4f\t%9.4f\t%9.4f\n',betapost11);

fprintf('\n');

%  95 percentile

qq22 = sort(deltam2);

betas22 = mean(deltam2);

m9522 = [qq22(ceil(store2*0.025),:);qq22(floor(store2*0.975),:)];

betapost22 = [qq22(ceil(store2*0.025),:);betas22;qq22(floor(store2*0.975),:)];

fprintf('\n');

disp('Posterior MCMC Estimates of gamma Coeffs with 95% interval');

fprintf('\n');

fprintf('lower\t\t     mean\t\t     upper\n');

fprintf('\n');

fprintf('%9.4f\t%9.4f\t%9.4f\n',betapost22);

fprintf('\n');

%  95 percentile

qq33 = sort(deltam3);

betas33 = mean(deltam3);

m9533 = [qq33(ceil(store2*0.025),:);qq33(floor(store2*0.975),:)];

betapost33 = [qq33(ceil(store2*0.025),:);betas33;qq33(floor(store2*0.975),:)];
```

```
fprintf('\n');

disp('Posterior MCMC Estimates of Beta Coeffs with 95% interval');

fprintf('\n');

fprintf('lower\t\t     mean\t\t     upper\n');

fprintf('\n');

fprintf('%9.4f\t%9.4f\t%9.4f\n',betapost33);

fprintf('\n');
```

4. Results from Bayesian Estimation

```
Vd Variance of the error terms in Hierarchical Regression


invVm =

    1.9826    0.4064    0.1660
    0.4064    0.8546    0.3458
    0.1660    0.3458    0.5512


ans =

Sigma: standard deviation of error term in Demand Model


ans =

   35.7277


ans =

Coefficients for etha (expected attractiveness


ans =

   -0.2124
    0.4755
   -0.8956
    0.4333
   -0.0513
   -0.6328
    0.5679
    0.9172


ans =

Coefficients for gamma (peak of attractiveness)
```

```
ans =

   -0.1750
   -0.0930
   -0.0511
    0.0892
    0.0071
   -0.1472
    0.4154
   -0.3421


ans =

Coefficients for beta (rate of attractiveness)


ans =

    0.0166
   -0.1052
    0.5231
    0.0129
    0.1386
   -0.4634
    0.2323
   -0.1455

Posterior MCMC Estimates of etha Coeffs with 95% interval

lower            mean              upper

  -0.2322     -0.2124     -0.1928
   0.4113      0.4755      0.5402
  -0.9634     -0.8956     -0.8265
   0.4099      0.4333      0.4558
  -0.0605     -0.0513     -0.0419
  -0.6970     -0.6328     -0.5701
   0.5262      0.5679      0.6116
   0.8996      0.9172      0.9337


Posterior MCMC Estimates of gamma Coeffs with 95% interval

lower            mean              upper

  -0.1955     -0.1750     -0.1552
  -0.1588     -0.0930     -0.0310
  -0.1204     -0.0511      0.0176
   0.0648      0.0892      0.1128
  -0.0023      0.0071      0.0161
  -0.2118     -0.1472     -0.0832
   0.3718      0.4154      0.4593
  -0.3592     -0.3421     -0.3251
```

Posterior MCMC Estimates of Beta Coeffs with 95% interval

lower	mean	upper
-0.0029	0.0166	0.0368
-0.1676	-0.1052	-0.0386
0.4545	0.5231	0.5925
-0.0113	0.0129	0.0371
0.1291	0.1386	0.1477
-0.5268	-0.4634	-0.3981
0.1866	0.2323	0.2767
-0.1629	-0.1455	-0.1276