# A Correlated Pareto/NBD Model

Peter S. Fader
www.petefader.com

Bruce G. S. Hardie[†]
www.brucehardie.com

January 2015

## 1 Introduction

Of all the assumptions associated with the Pareto/NBD model, the one that many researchers have the most problem with is the assumption that the transaction rate $\lambda$ and the "death" rate $\mu$ vary independently across customers. This is not nearly as restrictive as it may seem; more formally, we are assuming independent priors, which does not imply independence in the joint posterior distribution of $\Lambda$ and $M$. Nevertheless, there is a belief that this assumption is overly restrictive.

This note presents an informal description of a "correlated Pareto/NBD" model as (inelegantly) implemented in MATLAB[1] and applied to the CDNOW dataset. To accommodate correlation, we replace the Pareto/NBD assumption that heterogeneity in $\lambda$ and $\mu$ is captured by two independent gamma distributions with the assumption that the joint distribution of $\lambda$ and $\mu$ is a bivariate lognormal distribution, also known as the $S_{LL}$ distribution (Johnson 1949). (Strictly speaking, it is incorrect to call this model a correlated Pareto/NBD; it would be more correct to call it the $S_{LL}$-Exponential/Poisson model.)

We assume that the reader is familiar with the mathematics behind the Pareto/NBD model (e.g., Fader and Hardie 2005) as well as the mechanics of implementing it in MATLAB (Fader et al. 2005). We also assume that the reader is familiar with the concepts of the method of maximum simulated likelihood, the basics of evaluating integrals using Monte Carlo integration, the use of the Cholesky decomposition of the covariance matrix when generating draws from a multivariate normal distribution, and so on.



[1]The analysis was undertaken using R2012a (7.14.0.739), 64-bit (win64).

## 2 Parameter Estimation

Given the assumptions of the Pareto/NBD model, the individual-level likelihood function is

$$L(\lambda, \mu \,|\, x, t_x, T) = \frac{\lambda^x \mu}{\lambda + \mu} e^{-(\lambda+\mu)t_x} + \frac{\lambda^{x+1}}{\lambda + \mu} e^{-(\lambda+\mu)T} \,. \tag{1}$$

We assume that the joint distribution of $\Lambda$ and $M$ is bivariate lognormal,

$$\begin{bmatrix} \ln(\lambda) \\ \ln(\mu) \end{bmatrix} \sim MVN\left( \begin{bmatrix} \nu_\Lambda \\ \nu_M \end{bmatrix}, \begin{bmatrix} \sigma_\Lambda^2 & \sigma_{\Lambda M} \\ \sigma_{M\Lambda} & \sigma_M^2 \end{bmatrix} \right),$$

and therefore need to evaluate

$$L(\boldsymbol{\nu}, \boldsymbol{\Sigma} \,|\, x, t_x, n) = \int_0^\infty \int_0^\infty L(\lambda, \mu \,|\, x, t_x, n) f(\lambda, \mu \,|\, \boldsymbol{\nu}, \boldsymbol{\Sigma}) \, d\lambda \, d\mu \,.$$

There is no analytic solution to this double integral. We therefore evaluate it using Monte Carlo simulation; that is, we estimate the model parameters using the method of maximum simulated likelihood. We do this in the following manner in MATLAB.[2]

- The data are read into MATLAB using the following script:

```
% load_data.m
% Loads the CDNOW data from the spreadsheet cdnow_data.xls
global p1x p2x tx T
tmpdata = xlsread('d:\cdnow_data.xls','Individual-level Data','b2:e2358');
p1x = tmpdata(:,1);
tx = tmpdata(:,2);
T = tmpdata(:,3);
p2x = tmpdata(:,4);
clear tmpdata;
```

  (See Fader et al. (2005) for further details.)

- We generate two vectors of random numbers drawn from a normal distribution with mean zero and variance one using the following script. (For the purposes of this analysis, we are using 100,000 draws to evaluate the integrals of interest.)

```
% generate_common_random_numbers.m
global Z
rng('default');
Z = randn(100000,2);
```

[2]Note that our simulated likelihood routines are very crude and should not be viewed an example of good practice.

- We first consider an uncorrelated model. The following function computes the value of the sample log-likelihood function for a given set of model parameters (contained in the vector `param`):

```
function [f,g]= ll_uncorr(param)
% ll_uncorr.m -- evaluate the log-likelihood function for the
% "correlated Pareto/NBD model" with zero correlation.
global p1x tx T Z
param
% Part A
Nu = param(1:2);
Sigma = diag([param(3:4)]);
Y = Z*sqrt(Sigma);
lam = exp(Nu(1) + Y(:,1));
mu = exp(Nu(2) + Y(:,2));
% Part B
lammu = lam + mu;
lik = zeros(length(p1x),1);
for i = 1:length(p1x)
   tmp = lam.^(p1x(i))./lammu;
   lik(i) = mean(tmp.*(mu.*exp(-lammu*tx(i)) + lam.*exp(-lammu*T(i))));
end
f = -sum(log(lik))
g=[];
```

  The first and third elements of `param` are the mean and variance of the normal distribution whose random variates are transformed into values of $\lambda$ using the exponential transformation in "Part A". Similarly, the second and fourth elements of `param` are the mean and variance of the normal distribution whose random variates are transformed into values of $\mu$.

  Now that we have, in this case, 100,000 draws from the lognormal distributions for $\lambda$ and $\mu$, "Part B" sees us performing the Monte Carlo integration: computing (1) for each draw and taking the average of the resulting quantity. The log of this is the value of the (simulated) log-likelihood function for the parameter values contained in `param`.

- We use the `fmincon` routine (which is part of the Optimization Toolbox) to find the maximum of the log-likelihood function (or, more correctly, the minimum of $-$LL) via the following script:

```
% estimate_uncorr.m
lb = -10*ones(1,4);
lb(3) = .0001; lb(4)= .0001;
ub =  5*ones(1,4);
initial = .1*ones(1,4);
[ params_uncorr ll_uncorr exitflag_uncorr ] = ....
   fmincon('ll_uncorr',initial,[],[],[],[],lb,ub)
```

- `fmincon` terminates at

```
params_uncorr = -3.5136   -3.5038    1.2905    3.0160
ll_uncorr = 9.5735e+03
```

  These results are discussed below.

- We now run the correlated model. The only change we need to make to the log-likelihood code concerns the creation of `Y` given the two columns of uncorrelated normal random numbers contained in `Z`. Rather than directly estimating the covariance matrix of $\ln(\Lambda)$ and $\ln(M)$, we estimate the elements of the Cholesky decomposition of the covariance matrix.

```
function [f,g]= ll_corr(param)
% ll_corr.m -- evaluate the log-likelihood function for the
% "correlated Pareto/NBD" model.
global p1x tx T Z
param
% Part A
Nu = param(1:2);
R = [ param(3) param(4); 0 param(5) ];
Y = Z*R;
lam = exp(Nu(1) + Y(:,1));
mu = exp(Nu(2) + Y(:,2));
% Part B
lammu = lam + mu;
lik = zeros(length(p1x),1);
for i = 1:length(p1x)
   tmp = lam.^(p1x(i))./lammu;
   lik(i) = mean(tmp.*(mu.*exp(-lammu*tx(i)) + lam.*exp(-lammu*T(i))));
end
f = -sum(log(lik))
g=[];
```

- We find the maximum of the log-likelihood function via the following script:

```
% estimate_corr.m
lb = -10*ones(1,5);
lb(3) = .0001; lb(5)= .0001;
ub =  5*ones(1,5);
initial = .1*ones(1,5);
[ params_corr ll_corr exitflag_corr ] = ....
   fmincon('ll_corr',initial,[],[],[],[],lb,ub)
```

- `fmincon` terminates at

```
params_corr = -3.5166   -3.5833    1.1541    0.1307    1.7671
ll_corr = 9.5732e+03
```

- The last three elements of `params_corr` are the elements of the upper triangular matrix ($\mathbf{R}$) that is the Cholesky decomposition of the covariance matrix. Our estimate of $\mathbf{\Sigma}$ is computed as $\mathbf{R}'\mathbf{R}$:

```
R = [ params_corr(3) params_corr(4); 0 params_corr(5) ];
R'*R
```

  These results are discussed below.

With reference to Table 1, we see that relaxing the assumption of independence in the heterogeneity distributions for $\lambda$ and $\mu$ does not lead to a significant reduction in the value of the log-likelihood function. However, it is interesting to note that replacing the two gamma distributions of the Pareto/NBD model with independent lognormal distributions does result in a noticeable improvement in the value of the log-likelihood function.

The moments of the lognormal-based models in the $(\lambda, \mu)$ space are computed using the standard transformations: For $\big[\ln(X_1), \ln(X_2)\big]' \sim N(\boldsymbol{\nu}, \mathbf{\Sigma})$,

$$\begin{aligned} E(X_i) &= \exp\big(\nu_i + \tfrac{1}{2}\sigma_i^2\big)\,, \\ \mathrm{var}(X_i) &= \exp\big(2\nu_i + \sigma_i^2\big)\big\{\exp\big(\sigma_i^2\big) - 1\big\}\,, \text{ and} \\ \mathrm{corr}(X_1, X_2) &= \frac{\exp(\sigma_{ij}) - 1}{\sqrt{\big\{\exp\big(\sigma_1^2\big) - 1\big\}\big\{\exp\big(\sigma_2^2\big) - 1\big\}}}\,. \end{aligned}$$

We compare these estimates to those associated with the Pareto/NBD model in the lower part of Table 1.

Given the insignificant difference in fit between the two lognormal-based models, it is not surprising to see that the associated means and variances are very similar. It is interesting to note that three of the four estimates are quite different from those associated with the Pareto/NBD model.

To get a better sense of the estimated heterogeneity distributions, the following script (which assumes we have the Statistics Toolbox) creates the associated gamma and uncorrelated lognormal plots, which are given in Figure 1. (We do not show the correlated lognormal distributions as they are virtually indistinguishable from the uncorrelated lognormal distributions and only clutter the plots.)

```
% create_lam_mu_plots.m
m = cumsum(.0001*ones(1,10000));
%
density_lam_uncorr = pdf('logn',m,params_uncorr(1),params_uncorr(3));
density_mu_uncorr = pdf('logn',m,params_uncorr(2),params_uncorr(4));
%
r     =  0.553;
alpha = 10.578;
```

| | Pareto/NBD | Lognormal heterogeneity | |
|---|---|---|---|
| | | Uncorr | Corr |
| Parameter Estimates | | | |
| $\nu_\Lambda$ | | −3.5136 | −3.5166 |
| $\nu_M$ | | −3.5038 | −3.5833 |
| $\sigma^2_\Lambda$ | | 1.2905 | 1.3320 |
| $\sigma^2_M$ | | 3.0160 | 3.1397 |
| $\sigma_{\Lambda M}$ | | -- | 0.1509 |
| LL | −9,590.0 | −9,573.5 | −9,573.2 |
| Moments in the $(\lambda, \mu)$ space | | | |
| $E(\Lambda)$ | 0.0523 | 0.0568 | 0.0578 |
| var$(\Lambda)$ | 0.0049 | 0.0085 | 0.0093 |
| $E(M)$ | 0.0519 | 0.1359 | 0.1335 |
| var$(M)$ | 0.0045 | 0.3585 | 0.3940 |
| corr$(\Lambda, M)$ | -- | -- | 0.0207 |

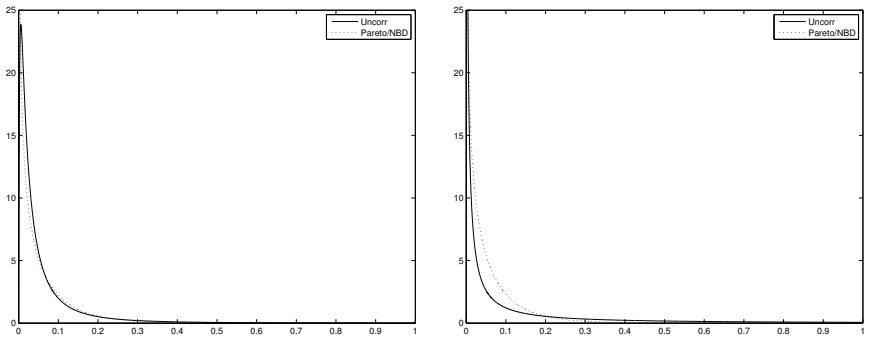
**Table 1:** Parameter estimates.

```
s     =  0.606;
beta  = 11.669;
density_lam_pnbd = alpha^r*m.^(r-1).*exp(-alpha*m)/gamma(r);
density_mu_pnbd = beta^s*m.^(s-1).*exp(-beta*m)/gamma(s);
%
plot(m,density_lam_uncorr,'k-',m,density_lam_pnbd,'k:');
axis([0,1,0,25]);
legend('Uncorr','Pareto/NBD');
print -depsc 'd:\dist_lam.eps'
%
plot(m,density_mu_uncorr,'k-',m,density_mu_pnbd,'k:');
axis([0,1,0,25]);
legend('Uncorr','Pareto/NBD');
print -depsc 'd:\dist_mu.eps'
```

We see that the lognormal distribution for heterogeneity in $\mu$ has a fatter tail than that of the associated gamma distribution. It is worth noting that the gamma pdf is monotone decreasing and has a vertical asymptote at zero when its shape parameter is less than one, as is the case for both of these gamma distributions. The lognormal can only have an interior mode, which is located $\exp(\nu - \sigma^2)$. (This is just visible in the distribution of $\lambda$ but not in that of $\mu$ due to the scaling of the plot axes.)

## 3 Assessing In-Sample Fit

To visually assess the fit of these three models, we compute the expected number of people making $0, 1, \ldots, 7+$ repeat purchases in the 39-week model




$\lambda$ $\mu$

**Figure 1:** Comparing the estimated distributions for heterogeneity in $\lambda$ and $\mu$.

calibration period for each of the three models and compare them to the actual frequency distribution.

The first thing we need to compute is $P(X(t) = x)$, the probability of making $x$ repeat transactions in the interval $(0, t]$. Conditional on $\lambda$ and $\mu$, we have

$$P(X(t) = x \mid \lambda, \mu) = \frac{(\lambda t)^x e^{-(\lambda+\mu)t}}{x!} + \frac{\lambda^x \mu}{(\lambda+\mu)^{x+1}} \left[ 1 - e^{-(\lambda+\mu)t} \sum_{i=0}^{x} \frac{\left[(\lambda+\mu)t\right]^i}{i!} \right].$$

For a randomly chosen individual,

$$P(X(t) = x \mid \boldsymbol{\nu}, \boldsymbol{\Sigma}) = \int_0^\infty \int_0^\infty P(X(t) = x \mid \lambda, \mu) f(\lambda, \mu \mid \boldsymbol{\nu}, \boldsymbol{\Sigma}) \, d\lambda \, d\mu \, .$$

The following MATLAB script evaluates the integrals for both the uncorrelated and correlated models using Monte Carlo simulation. (Note that the time period over which repeat transactions could have occurred varies across customers, depending on the day of their first purchase. Someone who made their first purchase at CDNOW on day $s$ of 1997 had $t = 39 - \frac{s}{7}$ weeks within which to make repeat purchases in the model calibration period.)

```
% compute_px.m
start_days = 84;
calib_weeks = 39;
t = calib_weeks - [1:start_days]'/7;
censor_point = 7;
% compute P(X(t)=x) for the uncorrelated model
Nu = params_uncorr(1:2);
Sigma = diag([params_uncorr(3:4)]);
Y = Z*sqrt(Sigma);
```

```
lam = exp(Nu(1) + Y(:,1));
mu = exp(Nu(2) + Y(:,2));
lammu = lam+mu;
px_uncorr = zeros(start_days,censor_point+1);
for y = 0:censor_point-1
   part1 = (lam*t').^y.*exp(-lammu*t')/prod(1:y);
   part2 = lam.^y.*mu./(lammu.^(y+1))*ones(1,start_days);
   part3 = 0;
   for j=0:y
      part3 = part3 + (lammu*t').^j/prod(1:j);
   end
   px_uncorr(:,y+1) = mean(part1 + part2.*(1-exp(-lammu*t').*part3))';
end
px_uncorr(:,censor_point+1) = 1 - sum(px_uncorr(:,1:censor_point),2);
% compute P(X(t)=x) for the correlated model
Nu = params_corr(1:2);
R = [ params_corr(3) params_corr(4); 0 params_corr(5) ];
Y = Z*R;
lam = exp(Nu(1) + Y(:,1));
mu = exp(Nu(2) + Y(:,2));
lammu = lam+mu;
px_corr = zeros(start_days,censor_point+1);
for y = 0:censor_point-1
   part1 = (lam*t').^y.*exp(-lammu*t')/prod(1:y);
   part2 = lam.^y.*mu./(lammu.^(y+1))*ones(1,start_days);
   part3 = 0;
   for j=0:y
      part3 = part3 + (lammu*t').^j/prod(1:j);
   end
   px_corr(:,y+1) = mean(part1 + part2.*(1-exp(-lammu*t').*part3))';
end
px_corr(:,censor_point+1) = 1 - sum(px_corr(:,1:censor_point),2);
```

Let $n_s$ be the number of customers who made their first purchase at CDNOW on day $s$ of 1997. The expected number of people in this cohort of new customers with $x$ repeat transactions is computed in the following manner:

$$E(f_x) = \sum_{s=1}^{84} n_s P(X(t - \tfrac{s}{7}) = x)\,,\ x = 0, 1, 2, \ldots$$

Assuming that the values of $P(X(t) = x)$ for the Pareto/NBD model are in the matrix `px_pnbd` (computed using the script `compute_px_pnbd.m`), the following script creates a plot of the three sets of expected frequencies along with the actual distribution of repeat transactions in the model calibration period:

```
% create_fit_distribution.m
% determine cohort size by day of trial
ns = zeros(1,start_days);
for i = 1:start_days;
   ns(i) = sum((T == (calib_weeks*7-i)/7));
```

```
end
%
freq_uncorr = ns*px_uncorr;
freq_corr = ns*px_corr;
freq_pnbd = ns*px_pnbd;
freq_act = zeros(1,censor_point+1);
for y = 0:censor_point-1
   freq_act(y+1) = length(find(p1x==y));
end;
freq_act(censor_point+1) = length(p1x) - sum(freq_act(1:censor_point));
%
bar([ freq_act' freq_uncorr' freq_corr' freq_pnbd' ])
legend('Actual','Uncorr','Corr','Pareto/NBD');
xlabel('# Transactions');
ylabel('Frequency');
label = [ ' 0'; ' 1';' 2'; ' 3'; ' 4'; ' 5'; ' 6'; '7+' ];
set(gca,'xticklabel',label);
colormap(gray);
print -depsc 'd:\fit_distribution.eps'
```

The resulting plot is presented in Figure 2. Consistent with the log-likelihood numbers, we note that the lognormal-based models provide a better fit to the data than the Pareto/NBD model.

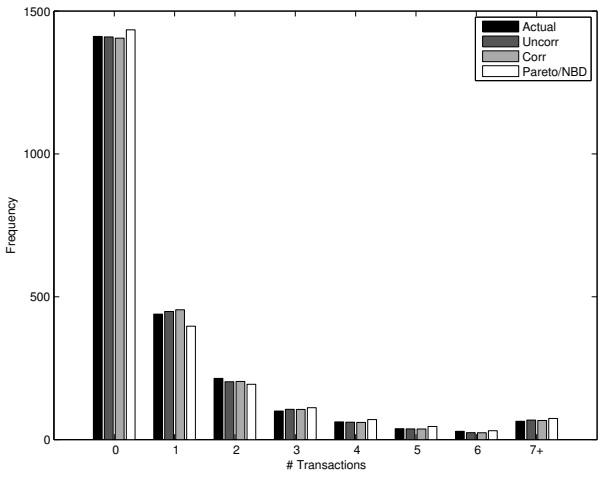



**Figure 2:** Predicted vs. actual frequency of repeat transactions in the model calibration period.

This visual assessment of relative fit is confirmed by computing the associated $\chi^2$ test statistics:

```
chisq_uncorr = sum((freq_act-freq_uncorr).^2./freq_uncorr)
chisq_corr = sum((freq_act-freq_corr).^2./freq_corr)
chisq_pnbd = sum((freq_act-freq_pnbd).^2./freq_pnbd)
```

giving us

```
chisq_uncorr = 2.3287
chisq_corr = 2.6202
chisq_pnbd = 11.9926
```

# 4 Assessing Out-of-Sample Aggregate Forecasts

Calibration-period fit is one thing. The big question is whether this improvement in model fit leads to any meaningful improvement in the associated predictions. One way to assess the performance of the models is to see how well the model-based predictions of repeat purchasing by the cohort of 2357 customers track the actual number of repeat transactions over time.

The first thing we need to compute is $E\big[(X(t)\big]$, the expected number of repeat transactions in the interval $(0, t]$. Conditional on $\lambda$ and $\mu$, this is given by

$$E\big[X(t) \,|\, \lambda, \mu\big] = \frac{\lambda}{\mu} - \frac{\lambda}{\mu} e^{-\mu t} \,. \tag{2}$$

For a randomly chosen individual,

$$E\big[X(t) = x \,|\, \boldsymbol{\nu}, \boldsymbol{\Sigma}\big] = \int_0^\infty \int_0^\infty E\big[X(t) \,|\, \lambda, \mu\big] f(\lambda, \mu \,|\, \boldsymbol{\nu}, \boldsymbol{\Sigma}) \, d\lambda \, d\mu \,.$$

We evaluate the integrals for both the uncorrelated and correlated models using the following MATLAB script:

```
% compute_ex.m
EndWk = 78;
EndDay = EndWk*7;
ex_uncorr = zeros(1,EndDay);
ex_corr = zeros(1,EndDay);
% compute E[(X(t)] for the uncorrelated model
Nu = params_uncorr(1:2);
Sigma = diag([params_uncorr(3:4)]);
Y = Z*sqrt(Sigma);
lam = exp(Nu(1) + Y(:,1));
mu = exp(Nu(2) + Y(:,2));
for i = 1:EndDay
   ex_uncorr(i) = mean((lam./mu).*(1-exp(-mu*i/7)));
end
% compute E[(X(t)] for the correlated model
Nu = params_corr(1:2);
R = [ params_corr(3) params_corr(4); 0 params_corr(5) ];
Y = Z*R;
lam = exp(Nu(1) + Y(:,1));
mu = exp(Nu(2) + Y(:,2));
for i = 1:EndDay
   ex_corr(i) = mean((lam./mu).*(1-exp(-mu*i/7)));
end
```

However, we are not interested in the expected number of repeat transactions for a randomly-chosen individual; rather we are interested in tracking

(and forecasting) the total number of repeat transactions by the cohort of customers. In computing this cohort-level number, we need to account for each individual's time of first purchase. The total cumulative number of repeat transactions is computed as follows:

$$\text{Total Repeat Transactions by } t = \sum_{s=1}^{84} \delta_{(t>\frac{s}{7})} n_s E\left[X(t-\tfrac{s}{7})\right],$$

where $\delta_{(t>\frac{s}{7})} = 1$ if $t > \frac{s}{7}$, 0 otherwise.

To compute the expected number of total repeat transactions for each of the 39 "calibration period" weeks and each of the following 39 "forecast period" weeks, we first compute this quantity for each of the $7 \times 78 = 546$ days and then extract every 7th number to yield the corresponding weekly numbers. Assuming that the values of $E\left[X(t)\right]$ for the Pareto/NBD model are in the vector `ex_pnbd` (computed using the script `compute_ex_pnbd.m`), this is computed via the following script, which also generates the associated tracking plot.

```
% create_cumulative_tracking_plot.m
start_days = 84;
calib_weeks = 39;
% determine cohort size by day of trial
ns = zeros(1,start_days);
for i = 1:start_days
   ns(i) = sum((T == (calib_weeks*7-i)/7));
end
%
tmpCumSls_uncorr = zeros(start_days,EndDay);
tmpCumSls_corr = zeros(start_days,EndDay);
tmpCumSls_pnbd = zeros(start_days,EndDay);
for i = 1:start_days
   tmpCumSls_uncorr(i,:) = [ zeros(1,i) ex_uncorr(1:EndDay-i) ];
   tmpCumSls_corr(i,:) = [ zeros(1,i) ex_corr(1:EndDay-i) ];
   tmpCumSls_pnbd(i,:) = [ zeros(1,i) ex_pnbd(1:EndDay-i) ];
end
DailySls_uncorr = ns*tmpCumSls_uncorr;
DailySls_corr = ns*tmpCumSls_corr;
DailySls_pnbd = ns*tmpCumSls_pnbd;
% extract weekly numbers
CumSls_uncorr = [];
CumSls_corr = [];
CumSls_pnbd = [];
for i = 1:EndWk
   CumSls_uncorr(i) = DailySls_uncorr(i*7);
   CumSls_corr(i) = DailySls_corr(i*7);
   CumSls_pnbd(i) = DailySls_pnbd(i*7);
end
% load actual cumulative repeat sales data
actual = xlsread('d:\cdnow_data','Cum. Repeat Sales','b1:b78');
% create tracking plot of cumulative repeat sales (pred. vs actual)
```

```
plot(1:EndWk,actual,'k', 1:EndWk,CumSls_uncorr,'k-.', ....
    1:EndWk,CumSls_corr,'k:', [39 39],[0 5000],'k--', ....
    1:EndWk,CumSls_pnbd,'k--');
axis([0,78,1,5000]);
set(gca,'YTick',[ 0 1000 2000 3000 4000 5000 ]);
set(gca,'XTick',[ 0 13 26 39 52 65 78 ]);
xlabel('Week'); ylabel('Cum. Rpt Transactions');
legend('Actual','Uncorr','Corr','Pareto/NBD',2);
print -depsc 'd:\cum_tracking.eps'
```

The resulting plot is presented in Figure 3. We note that the lognormal-based models appear to be slightly less accurate than the Pareto/NBD in tracking cumulative repeat purchasing out-of-sample.

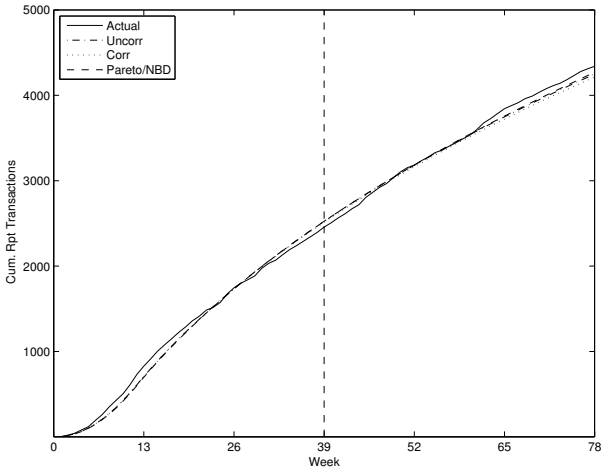



**Figure 3:** Comparing predicted vs. actual cumulative repeat transactions over time.

This visual assessment of relative tracking performance is confirmed by computing the out-of-sample MSE:

```
mse_uncorr = mean((actual(40:78)'-CumSls_uncorr(40:78)).^2)
mse_corr = mean((actual(40:78)'-CumSls_corr(40:78)).^2)
mse_pnbd = mean((actual(40:78)'-CumSls_pnbd(40:78)).^2)
```

giving us

```
mse_uncorr = 4.0139e+03
mse_corr = 6.5193e+03
mse_pnbd = 3.0958e+03
```

# 5 Conditional Expectations

Perhaps a more important evaluation of predictive performance considers the quality of the model predictions of future behavior conditional on past

behavior. In particular, we are interested in the expected number of transactions in the period $(T, T+t]$ by a customer with purchase history $(x, t_x, T)$; we denote this conditional expectation by $E\big[X(T, T+t) \,|\, , x, t_x, T\big]$.

Following the logic associated with the derivation of the Pareto/NBD conditional expectation, we obtain conditional expectation numbers for the lognormal-based models by evaluating the following expression:

$$\begin{aligned} E\big[X(T,&T+t) \,|\, \boldsymbol{\nu}, \boldsymbol{\Sigma}; x, t_x, T\big] \\ &= \int_0^\infty \int_0^\infty \Big\{ E\big[X(T, T+t) \,|\, \lambda, \mu, \Omega > T\big] \\ &\qquad \times P(\Omega > T \,|\, \lambda, \mu, x, t_x, T) \\ &\qquad\quad \times f(\lambda, \mu \,|\, \boldsymbol{\nu}, \boldsymbol{\Sigma}; x, t_x, T) \Big\} \, d\lambda \, d\mu \,. \end{aligned}$$

Noting that $E\big[X(T, T+t) \,|\, \lambda, \mu, \Omega > T\big]$ is the same as (2), recalling the basic Pareto/NBD model result,

$$P(\Omega > T \,|\, \lambda, \mu, x, t_x, T) = \frac{\lambda^x e^{-(\lambda+\mu)T}}{L(\lambda, \mu \,|\, x, t_x, T)} \,,$$

and noting that by Bayes' Theorem

$$f(\lambda, \mu \,|\, \boldsymbol{\nu}, \boldsymbol{\Sigma}; x, t_x, T) = \frac{L(\lambda, \mu \,|\, x, t_x, T) f(\lambda, \mu \,|\, \boldsymbol{\nu}, \boldsymbol{\Sigma})}{L(\boldsymbol{\nu}, \boldsymbol{\Sigma} \,|\, x, t_x, T)} \,,$$

this becomes

$$\begin{aligned} E\big[X(T,&T+t) \,|\, \boldsymbol{\nu}, \boldsymbol{\Sigma}; x, t_x, T\big] = \frac{1}{L(\boldsymbol{\nu}, \boldsymbol{\Sigma} \,|\, x, t_x, T)} \\ &\times \int_0^\infty \int_0^\infty \frac{\lambda^{x+1}}{\mu} e^{-(\lambda+\mu)T} \big(1 - e^{-\mu t}\big) f(\lambda, \mu \,|\, \boldsymbol{\nu}, \boldsymbol{\Sigma}) \, d\lambda \, d\mu \,. \end{aligned}$$

This is evaluated using the following MATLAB scripts for the uncorrelated and correlated models:

```
% compute_ce_uncorr.m
valid_weeks = 39; % period over while the ce is computed
Nu = params_uncorr(1:2);
Sigma = diag([params_uncorr(3:4)]);
Y = Z*sqrt(Sigma);
lam = exp(Nu(1) + Y(:,1));
mu = exp(Nu(2) + Y(:,2));
lammu = lam + mu;
lik = zeros(length(p1x),1);
ce_uncorr = zeros(length(p1x),1);
for i = 1:length(p1x)
   part1 = lam.^(p1x(i))./lammu;
   part2 = lam.*exp(-lammu*T(i));
   lik(i) = mean(part1.*(mu.*exp(-lammu*tx(i)) + part2));
   ce_uncorr(i) = mean(lam.^(p1x(i)+1)./mu.*exp(-lammu*T(i)) ....
      .*(1-exp(-mu*valid_weeks)))./lik(i);
end
```

```
% compute_ce_corr.m
valid_weeks = 39; % period over while the ce is computed
Mu = params_corr(1:2);
R = [ params_corr(3) params_corr(4); 0 params_corr(5) ];
Y = Z*R;
lam = exp(Mu(1) + Y(:,1));
mu = exp(Mu(2) + Y(:,2));
lammu = lam + mu;
lik = zeros(length(p1x),1);
ce_corr = zeros(length(p1x),1);
for i = 1:length(p1x)
   part1 = lam.^(p1x(i))./lammu;
   part2 = lam.*exp(-lammu*T(i));
   lik(i) = mean(part1.*(mu.*exp(-lammu*tx(i)) + part2));
   ce_corr(i) = mean(lam.^(p1x(i)+1)./mu.*exp(-lammu*T(i)) ....
      .*(1-exp(-mu*valid_weeks)))./lik(i);
end
```

Assuming that the corresponding Pareto/NBD conditional expectations are in the vector `ce_pnbd` (computed using the script `compute_ce_pnbd.m`), the following script plots the model-based conditional expectations along with the average of the actual number of transactions that took place in the 39-week forecast period, broken down by the number of calibration-period repeat purchases.

```
% create_ce_plot.m
ce_f_act = zeros(max(p1x)+1,1);
ce_f_uncorr = zeros(max(p1x)+1,1);
ce_f_corr = zeros(max(p1x)+1,1);
ce_f_pnbd = zeros(max(p1x)+1,1);
np1x = zeros(max(p1x)+1,1);
for y = unique(p1x)'
    isx = find(p1x==y);
    np1x(y+1) = length(isx);
    ce_f_act(y+1) = sum(p2x(isx))/np1x(y+1);
    ce_f_uncorr(y+1) = sum(ce_uncorr(isx))/np1x(y+1);
    ce_f_corr(y+1) = sum(ce_corr(isx))/np1x(y+1);
    ce_f_pnbd(y+1) = sum(ce_pnbd(isx))/np1x(y+1);
end
% create right-censored version for plot
censor = 7;  % right-censor at 7+
denom = sum(np1x(censor+1:length(np1x)));
ce_f_act_cen = ce_f_act(1:censor);
ce_f_act_cen(censor+1) = (np1x(censor+1:length(np1x))'...
    *ce_f_act(censor+1:length(np1x)))/denom;
ce_f_uncorr_cen = ce_f_uncorr(1:censor);
ce_f_uncorr_cen(censor+1) = (np1x(censor+1:length(np1x))'...
    *ce_f_uncorr(censor+1:length(np1x)))/denom;
ce_f_corr_cen = ce_f_corr(1:censor);
ce_f_corr_cen(censor+1) = (np1x(censor+1:length(np1x))'...
    *ce_f_corr(censor+1:length(np1x)))/denom;
ce_f_pnbd_cen = ce_f_pnbd(1:censor);
ce_f_pnbd_cen(censor+1) = (np1x(censor+1:length(np1x))'...
```

```
    *ce_f_pnbd(censor+1:length(np1x)))/denom;
%
plot([0:censor],ce_f_act_cen,'k',[0:censor],ce_f_uncorr_cen,'kx-.', ....
   [0:censor],ce_f_corr_cen,'k+--',[0:censor],ce_f_pnbd_cen,'kp:');
legend('Actual','Uncorr','Corr','Pareto/NBD',4);
xlabel('# Transactions in Weeks 1-39');
ylabel('Average # Transactions in Weeks 40-78');
axis([-.3 7.3 0 7]);
label = [ ' 0'; ' 1'; ' 2'; ' 3'; ' 4'; ' 5'; ' 6'; '7+' ];
set(gca,'xticklabel',label);
print -depsc 'd:\ce_plot.eps'
```

With reference to Figure 4, we see that the conditional expectations associated with the Pareto/NBD are more accurate than those from the lognormal-based models.

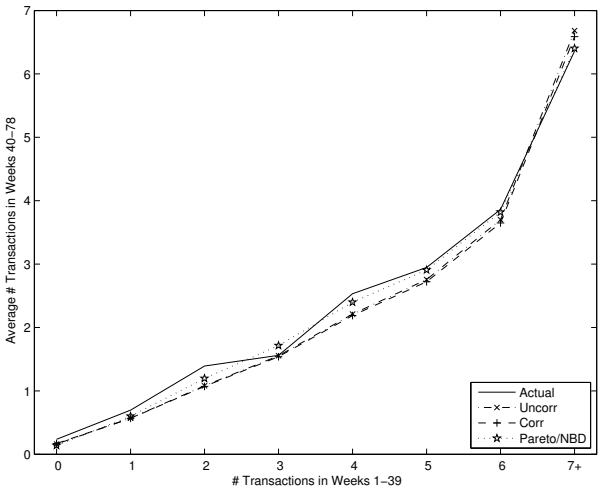



**Figure 4:** Predicted vs. actual conditional expectations of repeat transactions in the validation period as a function of calibration-period frequency.

# 6 Discussion

When dwelling on the Pareto/NBD model, some researchers are troubled by the assumption of independent gamma distributions. In this note, we have fitted a "correlated Pareto/NBD" model to the CDNOW dataset and compared the results to those of the Pareto/NBD model. We find that the lognormal-based models provide a better in-sample fit to the data, but do not find any significant correlation between the transaction rates ($\lambda$) and "death" rates ($\mu$). The original Pareto/NBD performs better than the lognormal-based models in the longitudinal out-of-sample analyses, especially in the conditional expectation analysis. (These results are broadly

consistent with the Bayesian analysis of the problem as presented in Abe (2009).)

In closing, it must be noted that the exact results of such a simulation-based analysis are sensitive to set of random numbers used to perform the Monte Carlo integration. Even though we have used a large number of draws, the results do change if another set of random numbers is used. However, the qualitative results remain the same: for the CDNOW dataset, the lognormal-based models provide a better in-sample fit, there is no evidence of correlation, and the Pareto/NBD performs better in out-of-sample analyses.

## References


Abe, Makoto (2009), ""Counting Your Customers" One by One: A Hierarchical Bayes Extension to the Pareto/NBD Model," *Marketing Science*, **28** (May–June), 541–553.

Fader, Peter S. and Bruce G. S. Hardie (2005), "A Note on Deriving the Pareto/NBD Model and Related Expressions." <`http://brucehardie.com/notes/009/`>

Fader, Peter S., Bruce G. S. Hardie, and Ka Lok Lee (2005), "A Note on Implementing the Pareto/NBD Model in MATLAB." <`http://brucehardie.com/notes/008/`>

Johnson, N. L. (1949), "Bivariate Distributions Based on Simple Translation Systems," *Biometrika*, **36** (3/4), 297–304.