Web Appendix A: WinBUGS code for the Proposed nested DP Model

```
model ndp {

for (i in 1:N)    {                    # N = 1511

# <--- for the calibration sample --->

lnvol[i]~         dnorm(mu[i],tau);

mu[i]        <-        a[1]*jan[i]        +a[2]*feb[i]        +a[3]*march[i]        +a[4]*april[i]
+a[5]*lnadex1[i]+a[6]*lnpacknum[i]

+a[7]*lnbotoz[i]+a[8]*bottle[i]+a[9]*light[i]+a[10]*amber[i]+a[11]*golden[i]+a[12]*lite[i]
          +b1[1, brand2[i],item2[i]] +b1[2, brand2[i],item2[i]]*lnprice1[i]
         +b1[3, brand2[i],item2[i]]*lndistbn[i] +b1[4, brand2[i],item2[i]]*lnpromo[i]
          +b1[5, brand2[i],item2[i]]*comptn.ndp0[i];

 errsq[i] <- pow(lnvol[i]-mu[i], 2);

 loglik[i] <- -0.5*log(6.283)+0.5*log(tau)-0.5*tau*pow(lnvol[i]-mu[i],2);

 dump[i]                                                                              <-
weeknum[i]+month[i]+jan[i]+feb[i]+march[i]+april[i]+lnpacknum[i]+lnbotoz[i]+bottle[i]
         +light[i]+amber[i]+golden[i]+ale[i]+lite[i]+reg[i]+lnprice1[i]+item2[i]+comptn.d
p0[i];

# <--- posterior predictive check - PPC --->
 lnvolnew[i] ~ dnorm(mu[i],tau);

                              } # i loop ends

# <--- fitted values from the prediction dataset --->
for (ip in 1:Np)          {                    # Np = 660

mup[ip] <- a[1]*janp[ip] +a[2]*febp[ip] +a[3]*marchp[ip] +a[4]*aprilp[ip]
          +a[5]*lnadex1p[ip]+a[6]*lnpacknump[ip] +a[7]*lnbotozp[ip]+a[8]*bottlep[ip]
         +a[9]*lightp[ip]+a[10]*amberp[ip]+a[11]*goldenp[ip]+a[12]*litep[ip]
          +b1[1, brand2p[ip],item2p[ip]] +b1[2, brand2p[ip],item2p[ip]]*lnprice1p[ip]
          +b1[3, brand2p[ip],item2p[ip]]*lndistbnp[ip]
          +b1[4, brand2p[ip],item2p[ip]]*lnpromop[ip]
         +b1[5, brand2p[ip],item2p[ip]]*comptn.ndp0p[ip];

errsqp[ip] <- pow(lnvolp[ip]-mup[ip], 2);

loglikp[ip] <- -0.5*log(6.283)+0.5*log(tau)-0.5*tau*pow(lnvolp[ip]-mup[ip],2);
```

```
dumpp[ip] <- weeknump[ip]+monthp[ip]+janp[ip]+febp[ip]+marchp[ip]+aprilp[ip]
          +lnpacknump[ip]+lnbotozp[ip]+bottlep[ip]+lightp[ip]+amberp[ip]
          +goldenp[ip]+alep[ip]+litep[ip]+regp[ip]+lnprice1p[ip]+item2p[ip]
          +comptn.dp0p[ip];

# PPC
 lnvolnewp[ip] ~ dnorm(mup[ip],tau)

                              } # ip loop ends



# < --- define priors --->

a[1:m] ~ dmnorm(nu.a[], Tau.a[1:m, 1:m])            # prior for fixed main effects a[]

for ( m1 in 1:m) { nu.a[m1]~ dnorm(0.001, 0.001) }

Tau.a[1:m,1:m] ~ dwish(RR[,],50)
for ( i in 1:m) {
           for ( j in (i+1):m) {  RR[i,j]<-0;                 RR[j,i]<-0      } # j loop ends
           RR[i,i]<- 0.25
                    } # i loop ends

tau ~ dgamma(0.001,0.001);   sig <- 1/tau;

# < ---  nDP code begins --->

# < --- let L, L1 be the truncation levels for the no. of cluster sat  the brand & SKU levels
# < --- First generate clustering parameters 'theta3' for  L x L1 values here:

 for (l1 in 1:L1) {
          rq[l1]~dbeta(1,alpha1);
          for (l in 1:L) {
                    r[l1, l] ~ dbeta(1, alpha);
                      theta3[l1, l, 1:d] ~ dmnorm(nu0[], Tau01[1:d, 1:d])         } # l loop ends
                    } # l1 loop ends


for ( k in 1:d) { nu0[k]~ dnorm(0.001, 0.001)          }        # k loop ends

Tau01[1:d,1:d]~dwish(RR[,],50)
for ( i in 1:d) {
           for ( j in (i+1):d) {  RR[i,j]<-0;                 RR[j,i]<-0      } # j ends
           RR[i,i]<- 0.25
```

```
            } # i ends

    # <--- Draw brand level cluster assignments for B=15 brands with an upper bound of L1 groups

     for (v in 1:B) {
            latent1[v,2] ~ dcat(q[1:L1]);
            for (l1 in 1:L1) {     Memb[v,l1] <- equals(latent1[v,2], l1) }
                } # v loop ends

    # <--- let data1 be a cross-list of brands & their SKUs (see data section below)
    # <--- Draw SKU level cluster assignments for the n=96 SKUs  --->

    for (m in 1:n) {
            latent2[m] ~ dcat(pi[latent1[data1[m,2],2],1:L])

           for (d1 in 1:d){
          b1[d1, data1[m,2], data1[m,1]] <- theta3[latent1[data1[m,2],2], latent2[m], d1] } # d1 ends

            for (l in 1:L) { Memb1[m,l] <- equals(latent2[m],l)  }
                } # m loop ends

    # <--- Preparing summary stats of how many clusters generated in each iteration
    # <--- of the Gibbs sampler: 'Tcluster' & 'Tcluster1'

     for ( j1 in 1:L) {
            Tmemb1[j1]<-sum(Memb1[,j1])
            Fmemb1[j1]<-step(Tmemb1[j1]-1)  } # j1 loop ends
     Tcluster1<-sum(Fmemb1[])                        # counts number of SKU clusters generated

     for ( j2 in 1:L1) {
            Tmemb[j2]<-sum(Memb[,j2])
            Fmemb[j2]<-step(Tmemb[j2]-1)      } # j2 loop ends
     Tcluster<-sum(Fmemb[])                          # counts no. of brand clusters generated

    # <--- set more priors & hyperpriors ---> #

     alpha1 ~ dgamma(a01, b01);
     a1 ~ dgamma(0.001, 0.001); b1 ~ dgamma(0.001, 0.001);
     tau1 ~ dgamma(2, 1)

    ### ----------------------------
    # <--- stick break for the two nested levels:
    ### ----------------------------
```

```
        q[1]<-rq[1]
        Rq[1] <- log(1-rq[1])

         for ( j in 2:(L1-1)) {
                          log(q[j]) <- log(rq[j]) + Rq[j-1]
                           Rq[j] <- Rq[j-1] + log(1-rq[j])      } # j loop ends

        q[L1]<-1-sum(q[1:(L1-1)])    # completes the classification into L1 clusters

        #  <--- set concentration parm priors and hyperpriors  --->#

         alpha0 ~ dgamma(a02, b02);            alpha <- 1+alpha0       #   to   avoid   computational
traps
        a02 ~ dgamma(0.001, 0.001); b02 ~ dgamma(0.001, 0.001);

         # <--- stick breaking L1 times for each of the L1 child DPs --->

        for (l1 in 1:L1) {
                        pi[l1,1] <- r[l1,1]
                        R1[l1,1] <- log(1-r[l1,1])

                         for ( j in 2:(L-1)) {
                                        log(pi[l1,j])<-log(r[l1,j])+ R1[l1,j-1]
                                         R1[l1,j] <- R1[l1,j-1] + log(1-r[l1,j])
        } # j ends

         pi[l1,L]<-1-sum(pi[l1,1:(L-1)])                          #   completes   assignment   into   L
clusters

              }       # ends 1:L1 loop

        #---------------- nDP part ends here -------------#

        # <--- summary metrics --->#

        # for PPCs
         s2new <- pow(sd(lnvolnew[]),2);      pval <- step(s2new - s2);
         s2newp <- pow(sd(lnvolnewp[]),2);   pvalp<- step(s2newp - s2p);

        # for non-normalized log-likelihoods

         llik <- sum(loglik[]);
         llikp <- sum(loglikp[])

         rmse <- sqrt(mean(errsq[]))
         rmsep <- sqrt(mean(errsqp[]))
```

```
        }       # model statement ends here.
### ------------------------------------------------------------ ###

DATA            # data section starts here

list(L=25,  L1=10,  B=15, n=96,  N=1511, Np=660,  d=5,

data1 = structure(.Data=c(1,1,
2,1,
3,1,
...
95,15,
96,15), .Dim=c(96, 2))                  ) # list ends

# DATA  calibration sample
weeknum[]       item2[]lnprice1[]       lndistbn[]      lnpromo[]       comptn.ndp0[]
comptn.dp0[]    brand2[]        lnpacknum[]     lnbotoz[]       bottle[]light[] amber[]
golden[]        ale[]   lite[]  reg[]   lnadex1[]       lnvol[] month[]         jan[]   feb[]
march[]         april[]
1       1       -2.323028707    -0.24098737     0.434385448     0.08966895      0.357309071
1       2.48490665      2.48490665      0       0       1       0       0       1       0
9.599540588     8.881836305     1       1       0       0       0
2       1       -2.319467689    -0.530938454    0.80317185      0.092378322     0.358247345
1       2.48490665      2.48490665      0       0       1       0       0       1       0
9.599540588     8.798454696     1       1       0       0       0
...
END

# DATA holdout Sample
<similar to above>


# INITIAL VALUES
list(tau=3, tau1=2, a01=2, a02=2, b01=1, b02=1, alpha1=4, alpha0=2,
 a=c(-0.2912,   -0.1733,   -0.1869,   -0.1041,   0.05896),     nu0=c(7.871,   -0.5988,   0.7654,
0.1158, -1)     )
```

Web Appendix B - Cluster Identification and Inference

First, some notation. Suppose there are n units to be clustered. Let z denote a n x 1 classification vector whose i^{th} component z_i stores the cluster label for unit i in any given draw from the Gibbs sampler. Let l=1,2..L be the nominal labels of the clusters that emerge from the analysis, some of which may be empty. For our application, *n* could be the number of brands (at the brand level) or the number of SKUs (at the SKU level). The algorithm can be summarized as follows:

Step 1: Build the posterior pairwise similarity matrix from the MCMC output of the model. The posterior pairwise similarity matrix (henceforth, PSM) is defined as a n x n matrix whose every cell (i, j) displays the probability π_{ij} conditional on data and parameters that units i and j come from the same cluster, thus:

$$\pi_{ij} = \Pr\left(z_i = z_j \mid .\right). \tag{B.1}$$

Note in Equation (B.1) that since the PSM is based on pairs of units co-occurring together, any relabeling of clusters does not affect the PSM.

Step 2: Maximize the posterior expected adjusted Rand index of the pairwise coincidence loss function to obtain the best partition z^{best}.

$$\underset{z^{\mathbf{best}}}{Min}\, E\left(\mid PSM\left(z^{best}\right) - PSM\left(z^{observed}\right)\mid\right). \tag{B.2}$$

While various methods have been proposed to partition units into clusters in the statistics literature using PSMs, Fritsch and Ickstadt (2009) develop a partitioning method based on maximizing the Posterior Expected Adjusted Rand (PEAR) index and compare it with extant benchmarks. We use PEAR to arrive at the best partitioning based on the data. As a robustness check, we compare the expected loss value with the approaches by Binder (1978) and Dahl (2006) following procedures outlined in Fritsch (2012).

Step 3: Collect cluster-specific parameters $\boldsymbol{\theta}_l$ for l=1,2..L and center them to get $\boldsymbol{\theta}_l^{*}$ by subtracting the corresponding overall sample mean $\bar{\boldsymbol{\theta}}$, thus:

$$\begin{aligned} \boldsymbol{\theta}_l &= \frac{1}{n_l} \sum_{i:z^{\mathbf{b}est}=l} \boldsymbol{\theta}_i, \;\; i = 1,2,..n; \\ \boldsymbol{\theta}_l^{*} &= \boldsymbol{\theta}_l - \bar{\boldsymbol{\theta}}, \qquad l = 1,2,..L. \end{aligned} \tag{B.3}$$

Here, n_l is the number of units belonging to cluster *l* under z^{best}.

Step 4: Compute summaries of $\boldsymbol{\theta}_{\mathbf{l}}^{*}$ and perform finite sample inference. By construction, the $\boldsymbol{\theta}_{\mathbf{l}}^{*}$ sum to zero. Finally, we compute useful summaries for the values $\boldsymbol{\theta}_{\mathbf{l}}^{*}$ by finding the Pr($\boldsymbol{\theta}_{\mathbf{l}}^{*} < 0$) or Pr($\boldsymbol{\theta}_{\mathbf{l}}^{*} > 0$). If Pr($\boldsymbol{\theta}_{\mathbf{l}}^{*} < 0$) or Pr($\boldsymbol{\theta}_{\mathbf{l}}^{*} > 0$) are above or below some threshold (typically 0.95 or 0.90), then we can infer that for cluster l, the group parameter values $\boldsymbol{\theta}_{\mathbf{l}}$ consistently cluster in a narrow region distinct from the mean for the sample as a whole.

Thus, we attempt to find if there exists a strong clustering "signal" in the data such that despite some variation in each iteration, on the whole, the clustering implied by the model is consistent with z^{best}. This post-processing approach represents a compromise between interpretable "hard clustering" groupings exemplified by z^{best}. and using model averaging techniques to assess the uncertainty involved in the parameter values corresponding to individual clusters (Molitor et al. 2010).

Web Appendix C: Robustness Checks on the Concentration Parameter Prior

Table 1A: Gamma Hyper-parameter specifics and Cluster Composition Overlaps

Panel A

Measure	Gamma(1,1)	Gamma(1,2)	Gamma(2,1)	Gamma(2,2)
Mean	1	2	2	4
Variance	1	4	2	8
Mode	0	0	1	2

Panel B

% of pairs that match	Gamma(1,1)	Gamma(1,2)	Gamma(2,1)	Gamma(2,2)
Gamma(1,1)				
Gamma(1,2)	93.38%			
Gamma(2,1)	89.22%	90.93%		
Gamma(2,2)	89.17%	90.71%	94.79%	
Gamma(a,b)	91.89%	94.48%	91.01%	91.41%

Table 2A: Summaries of Parameters of Interest from the 5 Models tested

Panel A	**Models with Different Priors on α at the brand & SKU levels**				
Measure	**Gamma(1,1)**	**Gamma(1,2)**	**Gamma(2,1)**	**Gamma(2,2)**	**Gamma(a,b)**
	Posterior Mean **[95% Credible Interval]**				
Brand level					
Concentration (α) parameter	2.79 [1.2, 5.19]	1.72 [0.57, 3.5]	1.43 [0.53, 2.79]	1.31 [0.48, 2.74]	1.55 [0.24, 3.91]
Observed #Clusters	4.38 [4, 5]	4.14 [4, 5]	5.04 [5, 6]	4.35 [4, 5]	4.19 [4, 5]
SKU level					
Concentration (α) parameter	7.57 [5.25, 10.2]	6.87 [4.73, 10.09]	6.39 [4.21, 9.51]	5.18 [3.57, 7.93]	11.66 [7.88, 16.23]
Observed #Clusters	21.66 [19, 24]	21.27 [18, 24]	22.55 [20, 25]	20.68 [18, 24]	23.06 [20, 25]
RMSE	0.18 [0.17, 0.18]	0.18 [0.17, 0.18]	0.18 [0.17, 0.18]	0.18 [0.17, 0.18]	0.18 [0.17, 0.18]
RMSE (holdout)	0.85 [0.75, 0.96]	0.82 [0.73, 0.94]	0.56 [0.51, 0.60]	0.72 [0.68, 0.78]	0.58 [0.54, 0.62]
Avg LogLikelihood	479.7 [452.7, 501.2]	489.2 [465, 511.9]	470.8 [444.3, 495]	473.8 [447.4, 498.1]	479.2 [452.9, 500.7]

Panel B	**Posterior Summaries of Random Parameter Estimates**				
Measure	**Gamma(1,1)**	**Gamma(1,2)**	**Gamma(2,1)**	**Gamma(2,2)**	**Gamma(a,b)**
	Average of Posterior Means (Std Deviation) **[Range]**				
Random Effect	9.69 (0.6) [8.66, 11.23]	9.76 (0.58) [8.72, 11.16]	9.42 (0.67) [8.15, 11.2]	9.43 (0.66) [8.06, 11.1]	9.6 (0.62) [8.5, 11.13]
Price Elasticity	-0.01 (0.04) [-0.11, 0.16]	-0.06 (0.04) [-0.13, 0.12]	-0.03 (0.04) [-0.13, 0.13]	-0.05 (0.03) [-0.12, 0.11]	-0.11 (0.05) [-0.24, 0.09]
Distribution Elasticity	0.86 (0.2) [0.23, 1.4]	0.91 (0.28) [0.18, 1.62]	0.81 (0.13) [0.27, 1.15]	0.8 (0.15) [0.28, 1.19]	0.84 (0.19) [0.26, 1.32]
Promotion Elasticity	0.12 (0.05) [0.03, 0.32]	0.12 (0.05) [0.02, 0.31]	0.12 (0.05) [0.02, 0.34]	0.12 (0.05) [0.03, 0.32]	0.12 (0.05) [0.03, 0.31]
Competition Response	-0.001 (0.03) [-0.095, 0.097]	0.001 (0.03) [-0.091, 0.101]	-0.003 (0.03) [-0.103, 0.096]	0.001 (0.03) [-0.098, 0.095]	-0.002 (0.03) [-0.113, 0.095]