Density Estimation Via Dirichlet Process Priors

Peter Rossi
GSB/U of Chicago

Density Estimation

Classical approaches:

Kernel Densities (choice of bandwidth, doesn't work in more than one or two dimensions. Requires a huge amount of data)

Series Expansion Methods (modify normal kernel) – don't work well if data density is really non-normal.

Finite Mixtures (MLEs are a mess) use slow EM. They are finite (you must promise to add mixture components). Limited support, etc.

Bayesian approach:

Compute predictive distribution of "next" observation.

Dirichlet Process Model: Two Interpretations

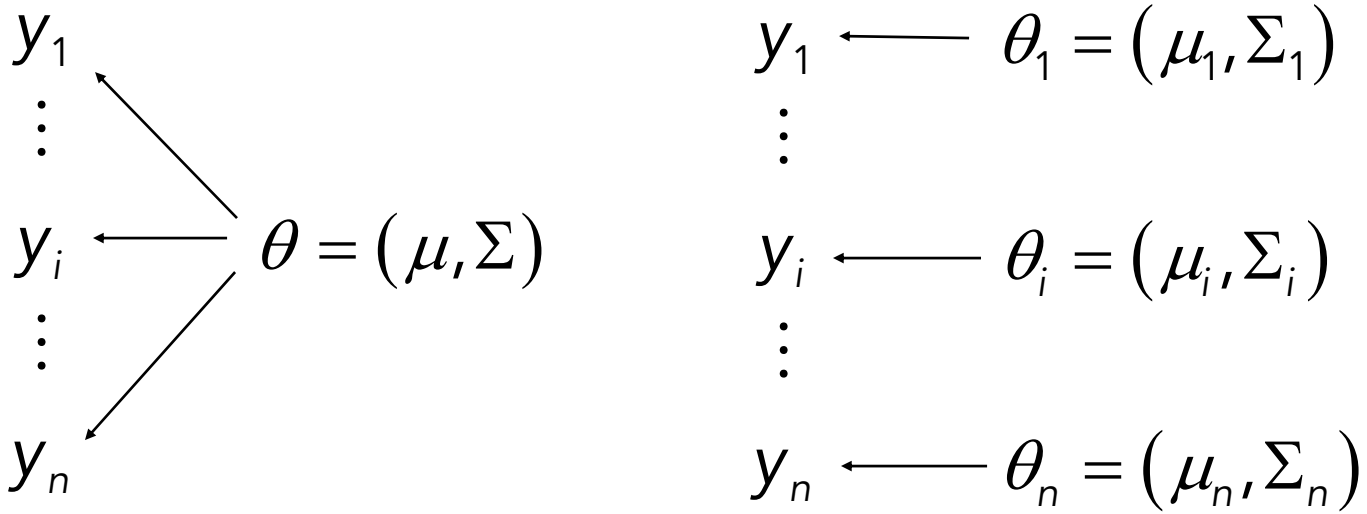
1. DP model is very much the same as a mixture of normals except we allow new components to be "born" and old components to "die" in our exploration of the posterior.
2. DP model is a generalization of a hierarchical model with a shrinkage prior that creates dependence or "clumping" of observations into groups, each with their own base distribution.

Ref: *Practical Nonparametric and Semiparametric Bayesian Statistics* (articles by West and Escobar/MacEachern)

Outline of DP Approach

We start from normal base (convenient but not required). How can we make distribution flexible?

Allow each obs to have its own set of parms:



Outline of DP Approach

This is a very flexible model that accomodates: non-normality via mixing.

However, it is not practical without a prior specification that ties the $\{ \theta_i \}$ together.

We need shrinkage or some sort of dependent prior to deal with proliferation of parameters (we can't literally have n independent sets of parameters).

Two ways: 1. make them correlated 2. "clump" them together by restricting to I* unique values.

Outline of DP Approach

Consider generic hierarchical situation:

$y_i \mid \theta_i$

$\theta_i \mid \lambda \sim G_0$

y are conditionally independent, e.g. normal with $\theta_i = (\mu_i, \Sigma_i)$

One component normal model: $\theta_i = (\mu, \Sigma)$

DAG:

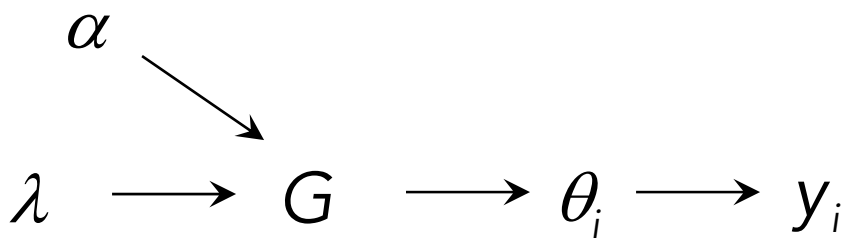
$$\lambda \longrightarrow \theta_i \longrightarrow y_i$$

Note: thetas are indep (conditional on lambda)

DP prior

Add another layer to hierarchy – DP prior for theta

DAG:

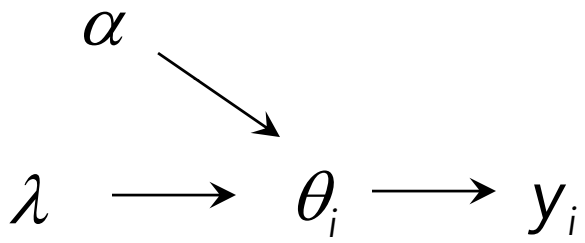


G is a Dirichlet Process – a distribution over other distributions. Each draw of G is a Dirichlet Distribution. G is centered on G_0 with tightness parameter α

DPM

Collapse the DAG by integrating out G

DAG:



$\{\theta_1, \ldots, \theta_n\}$ are now dependent with a mixture of DP distribution. Note: this distribution is not discrete unlike the DP. Puts positive probability on continuous distributions.

DPM: Drawing from Posterior

Basis for a Gibbs Sampler (so-called Polya Urn Rep):

$$\theta_j \left| Y, \theta_{-j} = \theta_j \right| y_j, \theta_{-j}$$

Why? Conditional Independence!

This is a simple update:

There are "n" models for θ_j each of the other values of theta and the base prior. This is very much like mixture of normals draw of indicators.

DPM: Drawing from Posterior

n models and prior probs:

$$\delta_i \qquad \text{with prior prob } \frac{1}{\alpha + (n-1)}$$ one of others

$$G_0(\lambda) \qquad \text{with prior prob } \frac{\alpha}{\alpha + (n-1)}$$ "birth"

$$\theta_j \left| \theta_{-j}, y_j, \lambda, \alpha \sim \begin{cases} q_0 & \theta_j \left| y_j, G_0(\lambda) \right. \\ q_i & \delta_i \quad i \neq j \end{cases} \right.$$

DPM: Drawing from Posterior

$$q_0 = p\left(M_0 \middle| y_j\right) = \int p\left(y_j \middle| \theta_j\right) p\left(\theta_j \middle| \lambda\right) d\theta_j \times p\left(M_0\right)$$

$$= \int p\left(y_j \middle| \theta_j\right) G_0\left(\theta_j \middle| \lambda\right) d\theta_j \times \frac{\alpha}{\alpha + (n-1)}$$

$$q_i = p\left(M_i \middle| y_j\right) = p\left(y_j \middle| \theta_i\right) \times \frac{1}{\alpha + (n-1)}$$

Likelihood Ratio. Like drawing indicators for FMN

Note: q need to be normalized! Conjugate priors can help to compute q_0.

DPM: Predictive Distributions or "Density Est"

$$p\left(y_{n+1}|y_1,\ldots,y_n\right)=\int p\left(y_{n+1}|\theta_{n+1}\right)p\left(\theta_{n+1}|y\right)d\theta_{n+1}$$

Note: this is like drawing from the first stage prior in hierarchical applications. We integrate out using the posterior distribution of the hyper-parameters.

$$p\left(\theta_{n+1}|Y\right)=\int p\left(\theta_{n+1}|\theta_1,\ldots,\theta_n\right)p\left(\theta_1,\ldots,\theta_n|y\right)d\theta_1\cdots d\theta_n$$

Both equations are derived by using conditional independence.

DPM: Predictive Distributions or "Density Est"

$$\theta_{n+1}|\theta \sim \begin{cases} \text{with prob } \dfrac{\alpha}{\alpha+n}, \text{ draw from } G_0(\lambda) \\ \text{with prob } \dfrac{1}{\alpha+n}, \text{ draw from } \delta_i \; i = 1,\ldots,n \end{cases}$$

Algorithm to construct predictive density:

1. draw $\theta_{n+1}|\theta^r, \lambda^r$
2. construct $p\left(y_{n+1}|\theta^r_{n+1}\right)$
3. average to obtain predictive density

Assessing the DP prior

Two Aspects of Prior:

α-- influences the number of unique values of θ

G_0 or λ -- govern distribution of proposed values of θ

e.g.

I can approximate a distribution with a large number of "small" normal components or a smaller number of "big" components.

Assessing the DP prior: choice of α

There is a relationship between α and the number of distinct theta values (viz number of normal components). Antoniak (74) gives this from MDP.

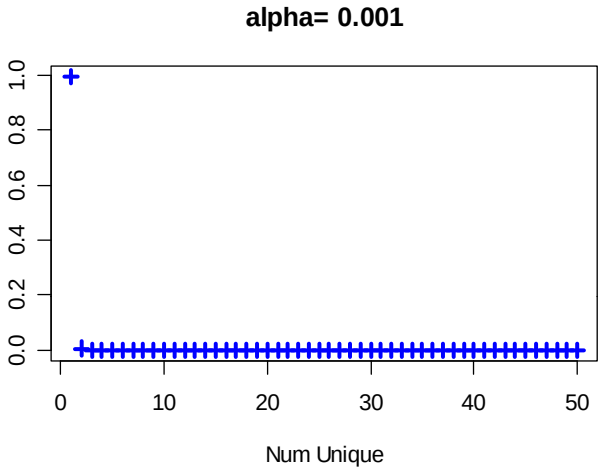
$$\Pr\left(l^* = k\right) = \left\|S_n^{(k)}\right\| \alpha^k \frac{\Gamma(\alpha)}{\Gamma(n+\alpha)}$$

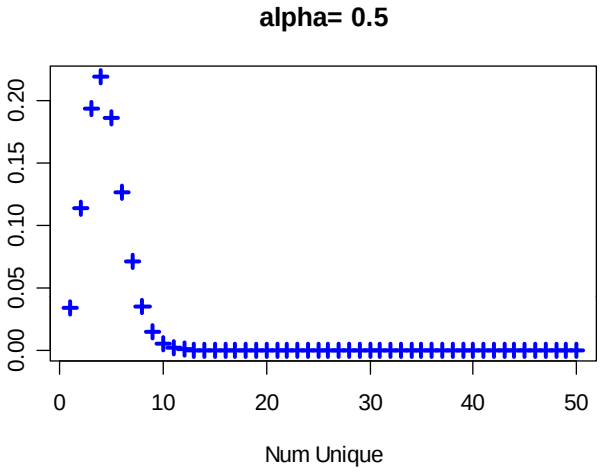
S are "Stirling numbers of First Kind." Note: S cannot be computed using standard recurrence relationship for n > 150 without overflow!

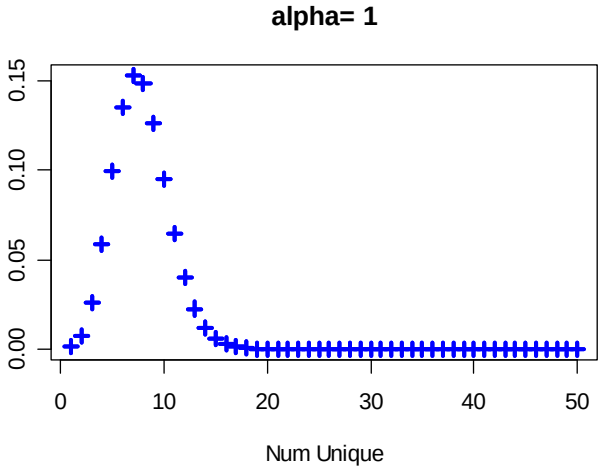
$$S_n^{(k)} \doteq \frac{\Gamma(n)}{\Gamma(k)} \left(\gamma + \ln(n)\right)^{k-1}$$

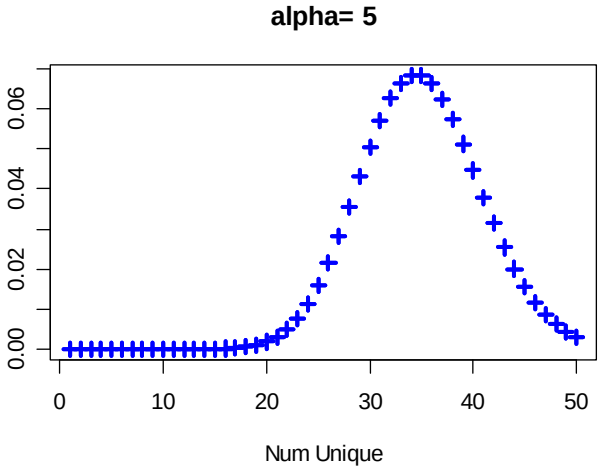
Assessing the DP prior: choice of α

For N=500









Assessing the DP prior: Priors on α

Fixing may not be reasonable. Prior on number of unique theta may be too tight.

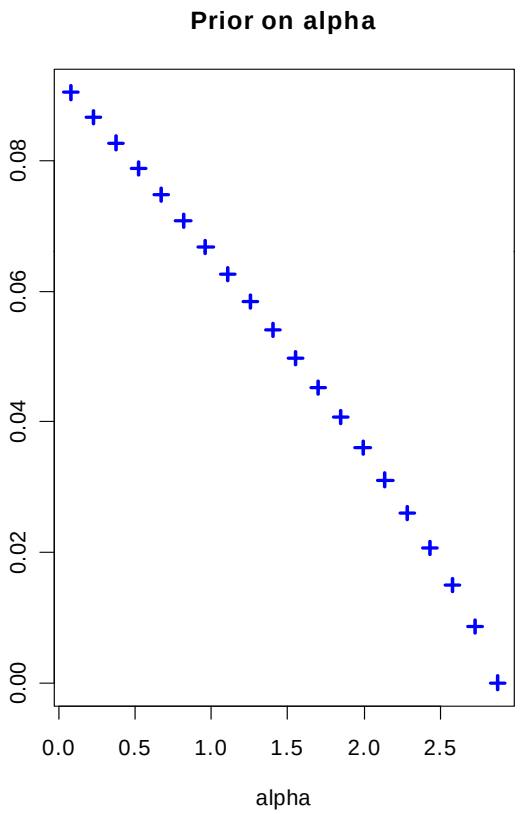
"Solution:" put a prior on alpha.

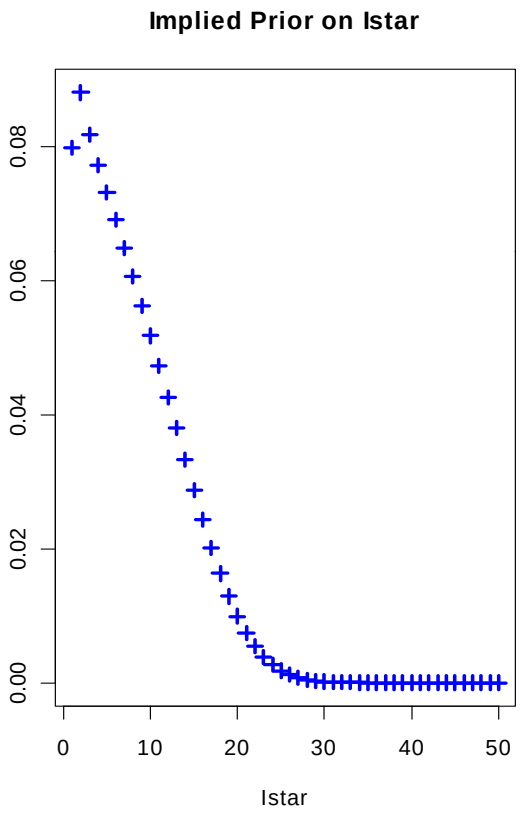
Assess prior by examining the priori distribution of number of unique theta.

$$p(l^*) = \int p(l^* | \alpha) p(\alpha) d\alpha$$

$$p(\alpha) \propto \left(1 - \frac{(\alpha - \underline{\alpha})}{(\bar{\alpha} - \underline{\alpha})} \right)^{\phi}$$

Assessing the DP prior: Priors on α





Assessing the DP prior: The Role of λ

$$q_0 = p\left(M_0 \middle| \varepsilon_j\right) = \int p\left(\varepsilon_j \middle| \theta_j\right) G_0\left(\theta_j | \lambda\right) d\theta_j \times \frac{\alpha}{\alpha + (n-1)}$$

Both α and λ determine the probability of a "birth."

Intuition:

1. Very diffuse settings of λ reduce model prob.
2. Tight priors centered away from y will also reduce model prob.

Must choose reasonable values. Shouldn't be very sensitive to this choice.

Putting a Prior on λ

$$G_0 : \mu \sim N\left(0, a^{-1}\Sigma\right); \Sigma \sim IW\left(\upsilon, \upsilon v I_k\right)$$

$$\text{mode}\left(\Sigma\right) = \frac{\upsilon}{\upsilon + 2} v I_k$$

Let's put a prior on υ, v, a. Note: here we are trying to let υ dictate tightness and v determine location of Σ. υ must be > dim for proper density.

$\left(a, \upsilon, v\right)$ *are indep*

$a \sim unif\left(a\text{lim}[1], a\text{lim}[2]\right)$

$v \sim unif\left(v\text{lim}[1], v\text{lim}[2]\right)$

$\upsilon = \dim\left(y\right) - 1 + \exp\left(z\right); z \sim unif\left(nu\text{lim}[1], nu\text{lim}[2]\right)$

$nu\text{lim}[1] > 0$

Coding DP in R

$\{\theta_i\}|Y,\lambda,\alpha$

Doesn't Vectorize

$\{\theta_i^*\}|ind,Y,\lambda$

"Remix" Step :
just like in FMN

$\alpha|I^*$

$\lambda|\{\theta_i^*\}$

Trivial (discrete)

or

$a|\{\theta_i^*\}$

$\upsilon|\{\theta_i^*\},v$

$v|\{\theta_i^*\},\upsilon$

q computations and conjugate draws are can be vectorized (if computed in advance for unique set of thetas).

Coding DP in R

$$\{\theta_i\}|Y,\lambda,\alpha$$

To draw indicators and new set of theta, we have to "Gibbs thru" each observation. We need routines to draw from the Base Prior and Posterior from "one obs" and base Prior (birth step).

We summarize each draw of using a list structure for the set of unique thetas and an indicator vector (length n).

We code the thetadraw in C but use R functions to draw from Base Posterior and evaluate densities at new theta value.

Coding DP in R: inside rDPGibbs

We must initialize theta, thetastar, lambda, alpha

```
for(rep in 1:R)
{
   n = length(theta)
   thetaNp1=NULL
   q0v = q0(y,lambda,eta)   # compute q0

   p=c(rep(1/(alpha+(n-1)),n-1),alpha/(alpha+(n-1)))

   nunique=length(thetaStar)

   ydenmat=matrix(double(maxuniq*n),ncol=n)
   ydenmat[1:nunique,]=yden(thetaStar,y,eta)
   #  ydenmat is a length(thetaStar) x n array of f(y[j,] | thetaStar[[i]]

   # use .Call to draw theta list
   out= .Call("thetadraw",y,ydenmat,indic,q0v,p,theta,lambda,eta=eta,
                  thetaD=thetaD,yden=yden,maxuniq,nunique,new.env())

         .
         .
         .
```

Coding DP in R: Inside thetadraw.C

```
/* start loop over observations */
   for(i=0;i < n; i++){
       probs[n-1]=NUMERIC_POINTER(q0v)[i]*NUMERIC_POINTER(p)[n-1];
       for(j=0;j < (n-1); j++){
             ii=indic[indmi[j]]; jj=i;      /* find element ydenmat(ii,jj+1) */
             index=jj*maxuniq+(ii-1);
             probs[j]=NUMERIC_POINTER(p)[j]*NUMERIC_POINTER(ydenmat)[index];
       }
       ind=rmultin(probs,n);

       if(ind == n){
          yrow=getrow(y,i,n,ncol);
          SETCADR(R_fc_thetaD,yrow);
          onetheta=eval(R_fc_thetaD,rho);
          SET_ELEMENT(theta,i,onetheta);
          SET_ELEMENT(thetaStar,nunique,onetheta);
          SET_ELEMENT(lofone,0,onetheta);
          SETCADR(R_fc_yden,lofone);
          newrow=eval(R_fc_yden,rho);
          for(j=0;j<n; j++)
          {NUMERIC_POINTER(ydenmat)[j*
               maxuniq+nunique]=NUMERIC_POINTER(newrow)[j];}
                     indic[i]=nunique+1;
                     nunique=nunique+1;}
       else {
          onetheta=VECTOR_ELT(theta,indmi[ind-1]);
          SET_ELEMENT(theta,i,onetheta);
          indic[i]=indic[indmi[ind-1]];
       }
   }
```

Call R functions to draw **theta**; compute new row of **ydenmat**

Draw new theta

Draw one of old thetas

theta is a R object (list of lists). This is a generalized vector.

All capitalized functions are defined in R header files. See .Call documentation for details.

Coding DP in R: inside rDPGibbs

```
  thetaStar=unique(theta);nunique=length(thetaStar)

#thetaNp1 and remix
  probs=double(nunique+1)
  for(j in 1:nunique) {
      ind = which(sapply(theta,identical,thetaStar[[j]]))
      probs[j]=length(ind)/(alpha+n)
      new_utheta=thetaD(y[ind,,drop=FALSE],lambda,eta)
      for(i in seq(along=ind)) {theta[[ind[i]]]=new_utheta}
      indic[ind]=j
      thetaStar[[j]]=new_utheta}
  probs[nunique+1]=alpha/(alpha+n)
  ind=rmultinomF(probs)
  if(ind==length(probs)) {
     thetaNp1=GD(lambda)}
  else {
     thetaNp1=thetaStar[[ind]]}

  # draw alpha
  alpha=alphaD(Prioralpha,nunique,gridsize=gridsize)

  # draw lambda
  lambda=lambdaD(lambda,thetaStar,alim=lambda_hyper$alim,
        nulim=lambda_hyper$nulim,vlim=lambda_hyper$vlim,gridsize=gridsize)
```

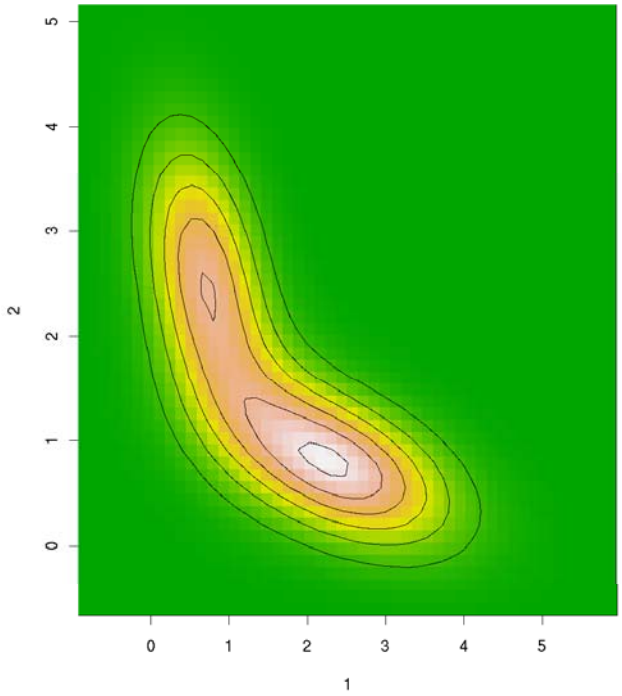
Example: Fit the "banana" density

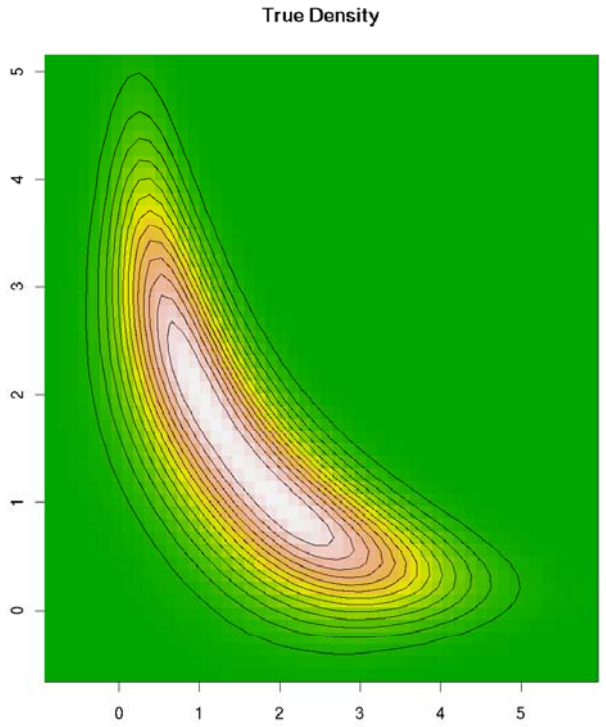
"banana" distribution of Meng and Barnard. Created by from conditional normals with nonlinear conditional means and variances. Simulate from using a Gibbs Sampler!

```
y2=banana(A=A,B=B,C1=C1,C2=C2,1000)

R=5000
Data2=list(y=y2)
Prioralpha=list(Istarmin=1,Istarmax=10,power=.8)
Prior2=list(Prioralpha=Prioralpha)
Mcmc=list(R=R,keep=1,maxuniq=200)

out2=rDPGibbs(Prior=Prior2,Data=Data2,Mcmc)

> names(out2)
[1] "alphadraw" "Istardraw" "adraw"     "nudraw"    "vdraw"     "nmix"
```

Example: Fit the "banana" density

"banana" distribution of Gelman and Meng.
Created by from conditional normals with nonlinear conditional means and variances.
Simulate from using a Gibbs Sampler!





Example: Fit the "banana" density

What's in nmix?

```
> names(out2$nmix)
[1] "probdraw" "zdraw" "compdraw"

> out2$nmix$compdraw[[1]]
[[1]]
[[1]]$mu
[1] 1.532062 1.649518

[[1]]$rooti
          [,1]     [,2]
[1,] 0.887829 0.961445
[2,] 0.000000 1.324959


> out2$nmix$probdraw[1]
[1] 1

> plot(out2$nmix)
```

compdraw is a list of list of lists:

compdraw[[r]][[1]][[1]]

compdraw[[r]][[1]][[2]]

plot invokes method, plot.bayesm.nmix

Finds marginals for each dimension and plots bivariates for each specified pair.

Averages marginals for each of R draws. mixDenBi or eMixMargDen. These are averages of ordinates of densities on a grid.