A Markov Bayesian Bootstrap, with Application (*?maybe?*) to Neural Differential Equations[1]

Version: 2023-04-13 by Michael and Vikas.

The setting Consider temporal data recordings $Z_1, Z_2, \cdots, Z_n$ residing in some low-dimensional Euclidean space (or maybe even a Riemannian manifold), and suppose we treat them as the realization of a Markov process governed by some unknown generative kernel

$$F(z, A) = P\left(Z_i \in A | Z_{i-1} = z\right)$$

for sets A and $i \geq 2$. Of course, if F did not really depend on z, then the data would correspond to `i.i.d.` draws from the unknown distribution F; but we are interested in an extension to Markov processes of Bayesian approximation schemes that are well-established in the `i.i.d.` case.

Random weighting in the i.i.d. case Recall first what happens in the `i.i.d.` case. Corresponding to the unknown generative distribution F are all sorts of features (e.g., moments, probabilities, etc), any one of which might be a target of inference from data $Z_1, \cdots, Z_n$. An interesting class of features are `model-guided` features, where we also have some (usually) parametric model

$$\mathbf{F}_\Theta = \left\{F_\theta : \theta \in \Theta\right\},$$

and say where each distribution F_θ admits a density f_θ with respect to a suitable dominating measure. This might be normals or finite mixtures of normals, for example. Whether the `true` generative F is in this class or not, we may still define the functional

$$\vartheta(F) = \arg\min_{\theta \in \Theta} E_F\left[-\log f_\theta(Z_i)\right].$$

In other words, we may identify a parameter value $\vartheta(F)$ in the finite-dimensional parameter space Θ with any generative F, as that point minimizing the Kullback Liebler divergence of F from the parametric model. More generally, we may identify parameter values relating F to any loss function l, and contemporary examples provide a slew of options (**citations**):

$$\vartheta(F) = \arg\min_{\theta \in \Theta} E_F l(\theta, Z_i). \tag{1}$$

A compelling approach to Bayesian inference (*uncertainty quantification*) is to treat the generative distribution F as fully unspecified, though with uncertainty governed by a Dirichlet process (DP) prior (**citations**). Then uncertainty about F continues to have a DP form in the posterior, reliant on both a prior measure $\alpha_0 F_0$ and the empirical measure $\mathbb{F}_n$ of the sample $Z_1, \cdots, Z_n$. This posterior *induces* a posterior distribution on the feature $\vartheta(F)$ by the functional mapping above. Notice that this UQ approach is quite different from the

[1]Technical Report no. supported in part by **

conventional model-based Bayesian inference, in which a prior distribution is constructed explicitly on $\mathbf{F}_\Theta$, and from which posterior inferences are derived, usually approximately, via Markov chain Monte Carlo or variational Bayes.

It may seem that we have just pushed the problem of UQ from model-based posterior analysis to some kind of nonparametric posterior analysis, and perhaps we are no closer to manageable computations in contemporary examples. However, it is now recognized that the posterior distribution of $\vartheta(F)$ is readily approximated by embarrassingly parallel *random weighting* calculations, which reformulate Don Rubin's Bayesian bootstrap procedure, and which apply to target features defined through optimization of an objective function (**citations**.)

Roughly speaking, the Bayesian bootstrap asks what distributions might have generated the observed data set (by contrast, frequentist bootstraps ask what other data sets might we observe on a hypothetical repeat of the experiment?). Having seen data $Z_1, \cdots, Z_n$, we know that F must have put some mass on each of those sample points; say it puts mass proportional to w_i on the ith data point. Then the expectation in~1 is proportional to

$$\sum_{i=1}^{n} w_i l(\theta, Z_i).$$

and the corresponding feature $\vartheta(F)$ is obtained by minimizing that weighted objective function. Interestingly, the Dirichlet Process (DP) posterior, in the limit when $\alpha_0 \longrightarrow 0$, converges weakly to a random distribution F supported on the sample points and equivalent to **random** weights w_i having a standard exponential distribution. Operationally, we repeatedly generate random weights $\mathbf{w} = (w_1, \cdots, w_n)$ and recompute $\vartheta(\mathbf{w})$. The empirical collection of $\vartheta(\mathbf{w})$ provides a basis for UQ on the target quantity. As an aside, empirical risk minimization solves for the single **point estimate**, $\vartheta(\mathbb{F}_n)$; the exponential weights used in $\vartheta(\mathbf{w})$ constitute a uniform distribution over the simplex of probability vectors supported on the sample, thus including, but of course greatly elaborating upon the single $\mathbb{F}_n$.

An aside on urns The probabilistic machinery through which the Dirichlet Process (DP) calculations are developed rests in part on exchangeability assumptions, reinforcement learning, predictive modeling, and urn sampling. We refer the reader to **cite** for a lucid development, and we note for our subsequent purposes with Markov processes that a simple urn-sampling metaphor is helpful. Namely, we suppose that prior to data being in hand, there is an urn U containing a measure $\alpha_0 F_0$; roughly, this measure represents the analyst's uncertainty about any data point Z_i, prior to observation (*the urn is metaphorical; it is one and the same thing as the measure; the total mass of the measure is like the number of balls in the urn*). After seeing Z_1, the analyst's mind updates – specifically the measure goes from $\alpha_0 F_0$ to, $U \leftarrow \alpha_0 F_0 + \delta_{Z_1}$, by adding a point mass at Z_1. Subsequent measurements continue to add mass, until the urn contains measure $U \leftarrow \alpha_0 F_0 + n\mathbb{F}_n$ by the end of data collection. This measure, as it turns out, is the centering measure of the posterior DP. Random weighting emerges, a posteriori, by sending $\alpha_0 \longrightarrow 0$ to eliminate F_0. (Note that urn sampling is like drawing a value from normalized probability distribution residing within the urn; the measure-adding reinforcement conveys how DP analysts learn from data.) (*comment on how weights can be generated from the fixed urn at the end of data collection*)

The Markov case We are not aware of other attempts to extend the Bayesian bootstrap to the Markov case. (*maybe some cites on partial exchangeability and finite state chains*). Instead of a single urn with a single measure $\alpha_0 F_0$, as in the i.i.d. case, let us suppose that every point z in the space is associated with a measure-containing urn, say U_z, prior to any data coming into view. All these urns will update during data collection. In the absence of any regularizing assumptions about the kernel $F(z, A)$, we might be obliged to update the urns as follows. Say with Z_1 fixed, let's update urn U_{Z_1} from its original state by adding point mass δ_{Z_2}; likewise at any transition from Z_i to Z_{i+1}, update the urn U_{Z_i} from its prior state by adding point mass at Z_{i+1}. For continuous recordings, this approach doesn't get us too far, even after all data are in hand, since we may expect to see single point masses added to $n-1$ of the continuum of urns. And then any limits we take on α_0 will leave us with very little empirical variation (probably just one single trajectory is realizable).

Regularizing assumption: Suppose that nearby urns (i.e., urns at nearby state space values z and z') tend to be similar, and, further, that all urn measures share the same support.

The common-support assumption is helpful, since it means that by the end of data collection all urns will include measures having some point mass at each of the observations; subsequent removal of the prior urn measures by sending α_0 to zero will mean that all urns will have the same set of support points. But the interesting thing is that the masses will not be the same over these support points, owing to the first part of the regularizing assumption. We encode the assumption using the idea of an envelope function, which in experiments we take as:

$$e(z, z') = \exp\{-d(z, z')/\tau\}$$

where d is a distance metric on the space and $\tau > 0$ is a scale parameter. We use this envelope to define how much mass to add to the different urns, after having observed data point Z_{i+1} emmitted after point Z_i:

$$U_z \longleftarrow U_z + e(z, Z_i)\delta_{Z_{i+1}}.$$

Thus, urns corresponding to `from` positions z far from the emitting point Z_i are given a low value of the envelope, and a relatively low mass at the support point. Such transitions have greatest effect on urns near to Z_i, and thereby locally reinforce learning about the transition kernel.

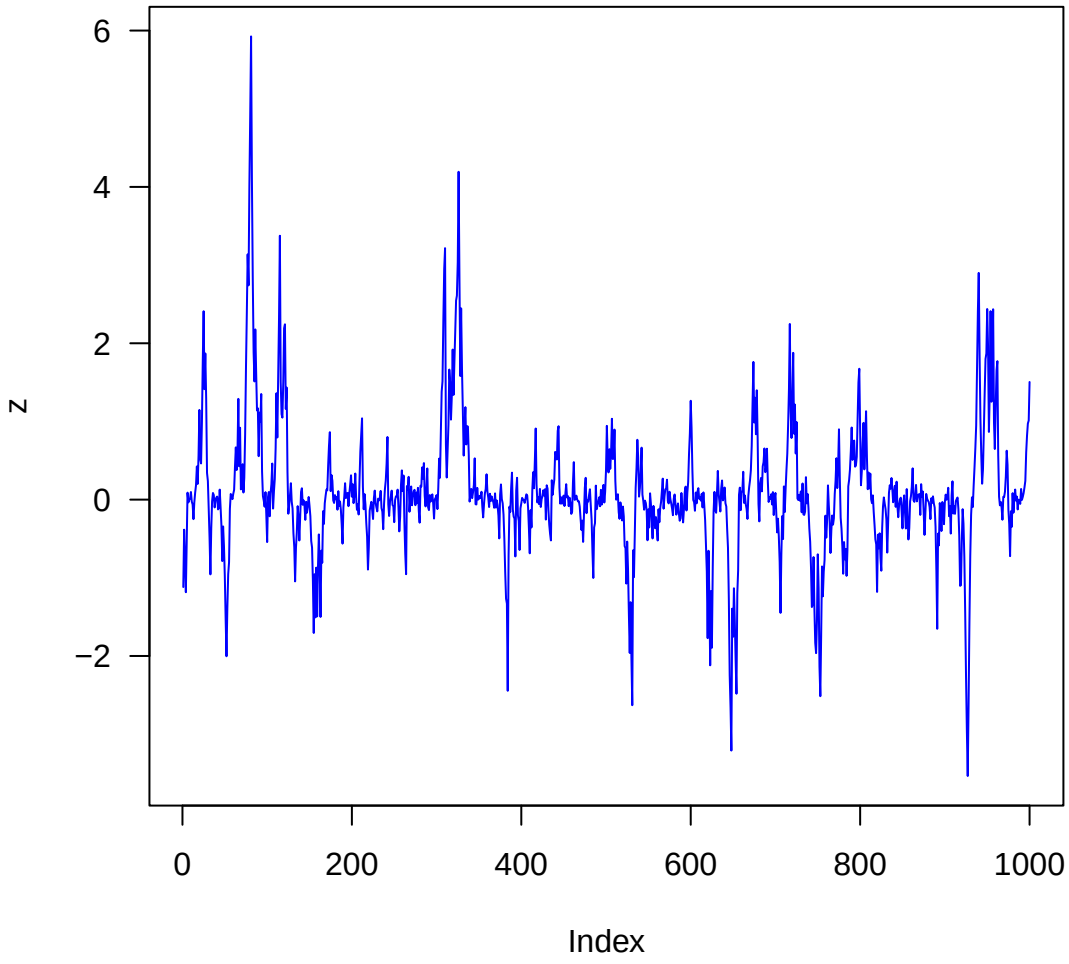
Conjecture: Letting n get large and for sufficiently small τ, the normalized urns will converge to the generative kernel F.

Synthetic example Here's some `R` code to simulate a synthetic auto-regressive model.

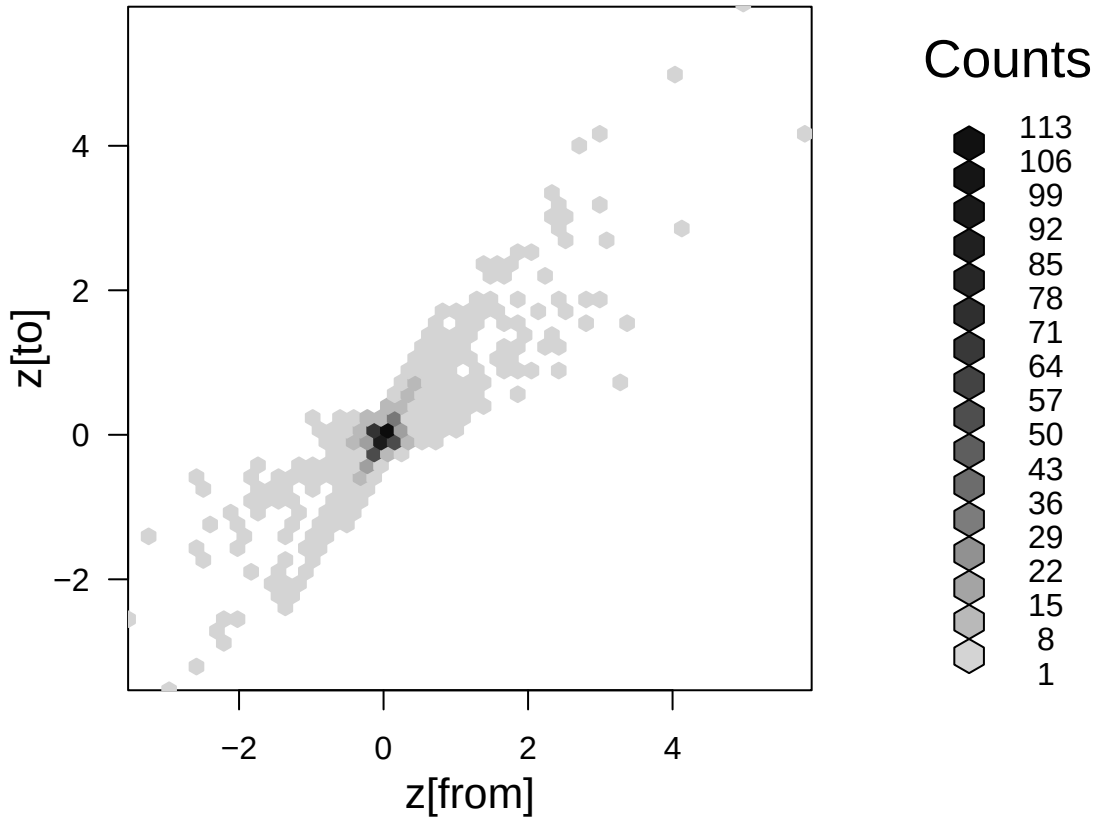
```
# a synthetic autoregressive example, but with goofy bimodal conditionals

set.seed(312345126)

rbimod <- function(mu, delta=2, sig=(1/4), phi=.95 )
 {
   # make a biomodal draw centered at mu
```

```
   sgn <- sample( c(-delta,delta), size=1, prob=c(1/2,1/2) )
   x <- phi*mu + sig*(rnorm(1)+sgn)*sqrt(abs(mu))
   x
 }

n <- 1000
x <- numeric(n)
x[1] <- rnorm(1)
for( i in 2:n){ x[i] <- rbimod(x[i-1]) }
```

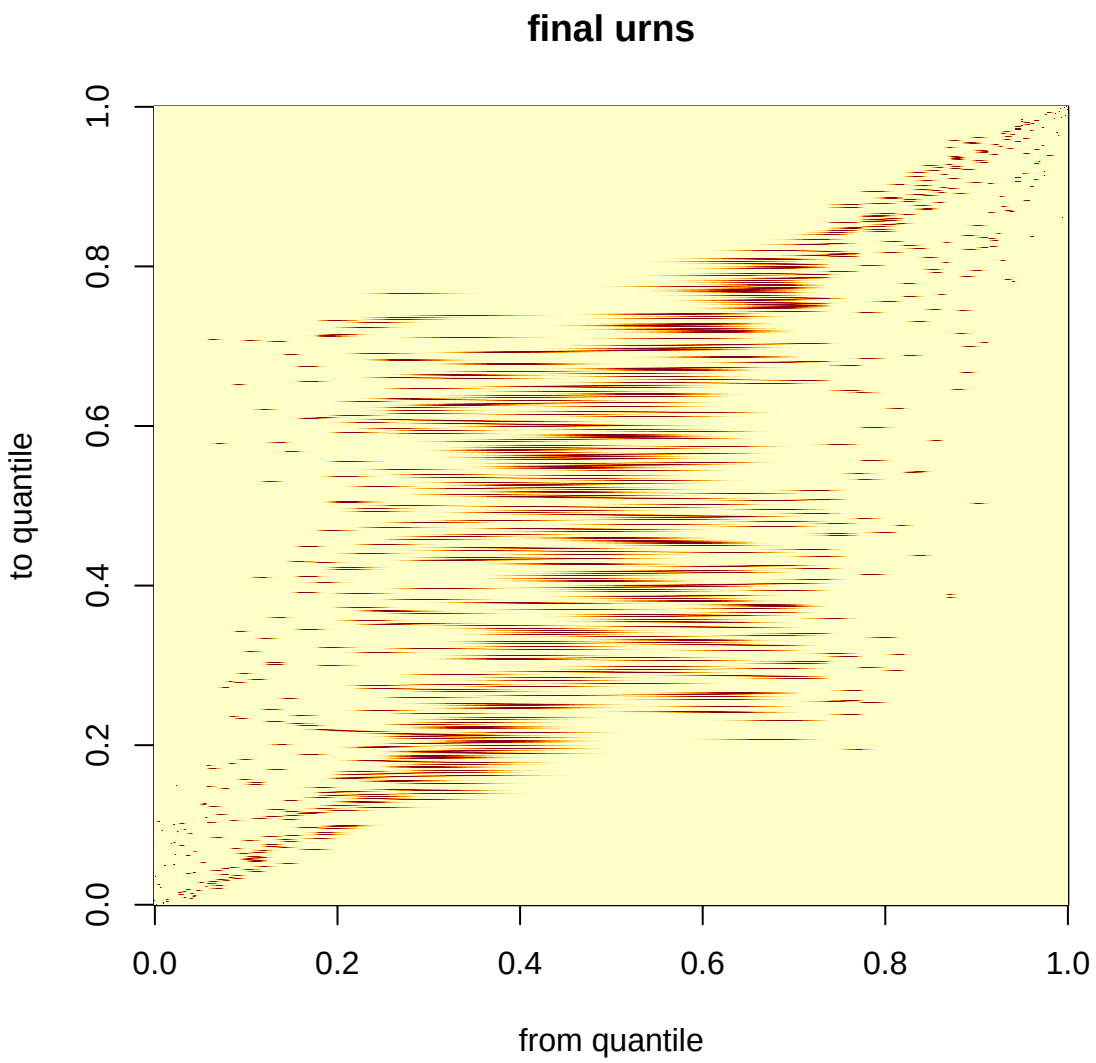
The plot below shows the realization of an artificial real-valued Markov process over $n = 1000$ steps.



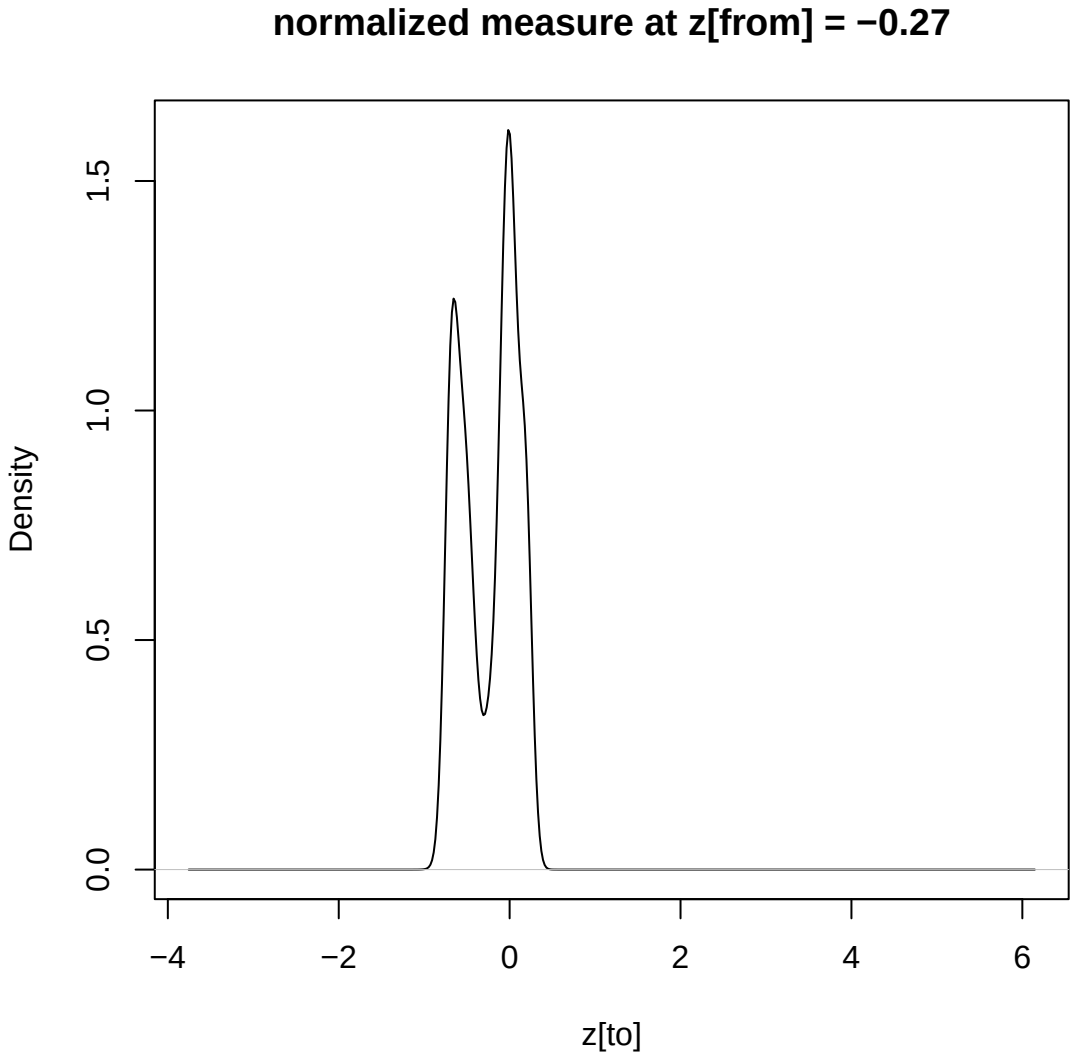
The lag plot hints at bimodality in the conditionals.



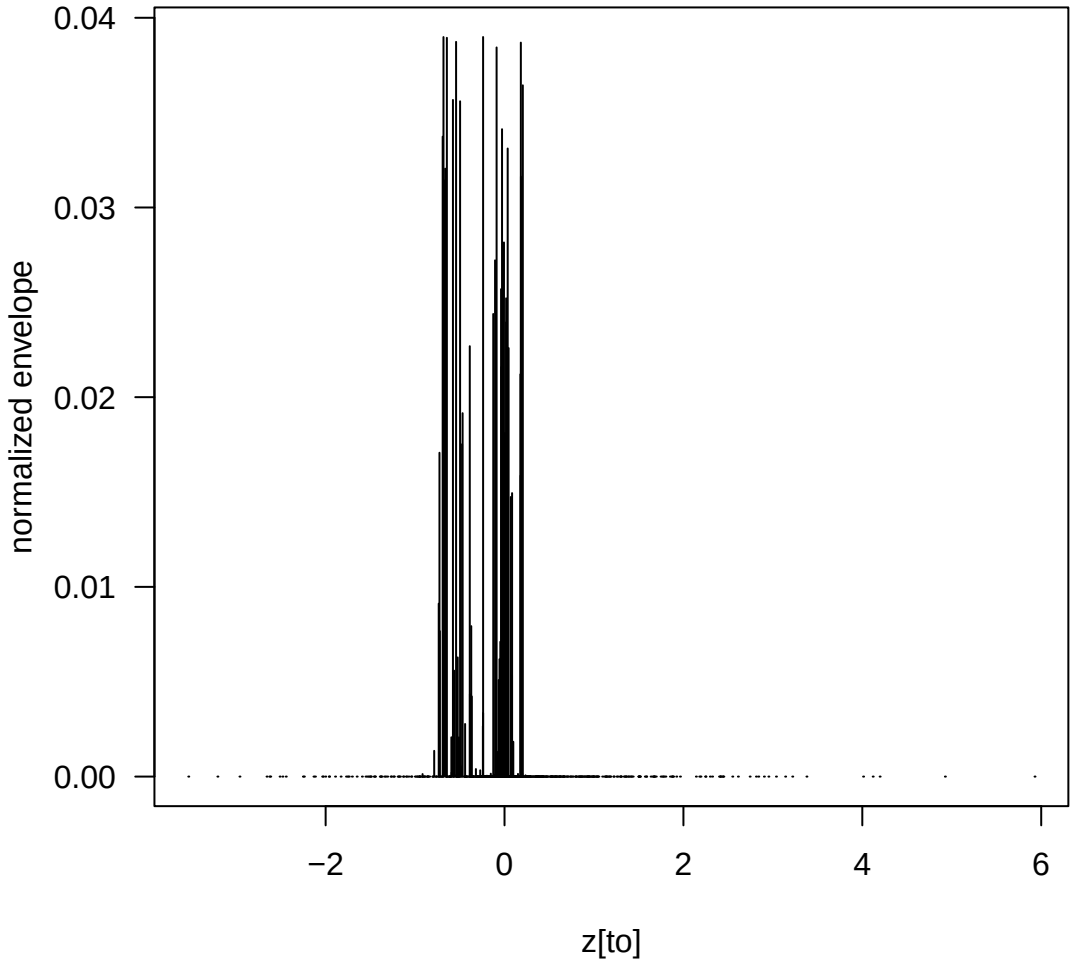
We deployed the envelope cacluation with $\tau = 10^{-3}$, and the figure shows a q-q version of the resulting kernel (which shows all the urns, after each is normalized to be a probability measure).



The plot below shows one conditional distribution (smoothed) from normalizing urn U_z where $z = -0.27$ is the 200^{th} order statistic, which looks about right considering the generative mixture of normals.



It's helpful to remember that we really have a discrete distributional estimate of the urn $U_{-0.27}$, which was smoothed into the displayed kernel density estimate above, but which has uneven masses according to the envelope, as below.



Thinking about the urn-measure representation, we would have something like this for the continuum of urns after sampling, and after limiting to remove the prior mass

$$U_z = \sum_{i=2}^{n} \delta_{Z_i}\, e(Z_{i-1}, z)$$

Markov Bayesian Bootstrap *how to make the matrix of weights*

identify a target feature; e.g. correlation, or some elaborate conditional probability

‘‘‘