The BMAmevt package: Bayesian Model Averaging at work for Multivariate Extremes

Anne Sabourin

sabourin@math.univ-lyon1.fr
anne.sabourin@lsce.ipsl.fr
annesab1@gmail.com

Université de Lyon I , France
Laboratoire des Sciences du Climat et de l'Environnement, Saclay, France.

1 Overview

This package is a 'Bayesian Model Averaging' (BMA) toolkit. The main purpose is the estimation of the dependence structure between the largest values of multivariate data, using the probabilistic framework of Multivariate Extreme Value Theory. However, the principal functions implemented here (MC MC sampler and simple Monte-Carlo sampler for the marginal model likelihoods, see below) could be used in any other statistical context.

This package has been developped to implement the methods proposed in [6].

The main functions are

- `posteriorMCMC`: A generic MC MC sampler (Metropolis-Hastings algorithm) to estimate the posterior distributions in individual models,
- `marginal.lkl` : A generic Monte-Carlo sampler to estimate the model likelihoods.

Plotting tools are also provided for the three dimensional case (functions `dgridplot, discretize`) to display contour lines and level sets of angular measures.

Any parametric model may be passed as argument. To 'plug' one's favorite model into the package, one only needs to define

- The likelihood function of the model (a parametric density on the simplex or any other finite dimensional sample space),
- The prior parameter distribution together with a prior sampler.

- The proposal distribution and simulating rule for the Metropolis-Hastings algorithm.

By way of example, two parametric models are pre-implemented: the Pairwise Beta (PB) model , valid in arbitrary " moderate" dimension[1], and a specific 'Nested Asymmetric' extension of the logistic model [4]), only valid with three dimensional data, which we refer to as the NL model. This nested extension was cited as an example in [7] and [2]. See [6] for more details.

2 Tutorial

We give here two basic examples showing how to use the functions of the package.

2.1 Simulated data

We consider the following problem: Given an angular dataset, how to conduct the estimation in the PB and NL model, and how to merge the estimated spectral measures ?

First, simulate the data:

```
> PBpar=c(0.7,3.1,0.45,2)
> NLpar=c(0.7,0.8)
> set.seed(1)
> mixDat=rbind(rpairbeta(n=50,dimData=3, par=PBpar),
+   rnestlog(n=50,par=NLpar))
```

Now, check that the two models have comparable posterior weights

```
> pWei=posteriorWeights (dat=mixDat,
+                   HparList=list( pb.Hpar, nl.Hpar ),
+                   lklList=list( dpairbeta, dnestlog ),
+                   priorList=list( prior.pb,  prior.nl ),
+                   priorweights=c(0.5,0.5),
+                   Nsim=50e+3,
+                   Nsim.min=10e+3, precision=0.1,
+                   show.progress=c(),
+                   displ=FALSE)
> pWei
```

You will obtain a matrix with first column (the posterior weights) approximately equal to `c(0.31,0.69)`.

Now, conduct the inference in each model, separately:

[1]It has been tested on five dimensional data sets in [3]. It would be computationally unrealistic to implement our methods to much higher dimensions.

```
> PBpost=posteriorMCMC.pb(dat=mixDat,Nsim=15e+3,Nbin=5e+3,
+   show.progress= c(100, 10e+3))
> NLpost=posteriorMCMC.nl(dat=mixDat,Nsim=15e+3,Nbin=5e+3,
+   show.progress= c(100, 10e+3))
```

It is recommended to check convergence:

```
> library(coda)
> heidel.diag(PBpost$stored.vals)
> heidel.diag(NLpost$stored.vals)
```

Have a look at the posterior predictive spectral densities:

- In the PB model only:

```
>   dev.new()
> PBpred=posterior.predictive.pb(post.sample=PBpost,from=NULL, to=NULL,
+   lag=40,npoints=60,eps=1e-3, equi=TRUE, displ=TRUE, main="PB predictive")
```

- In the NL model only:

```
> dev.new()
> NLpred=posterior.predictive.nl(post.sample=NLpost,from=NULL, to=NULL,
+   lag=40,npoints=60,eps=1e-3, equi=TRUE, displ=TRUE, main="NL predictive")
```

- Finally, the BMA predictive is:

```
> dev.new()
> dgridplot(0.31*PBpred + 0.69*NLpred, npoints=60, eps=1e-3, equi=TRUE,
+      main="BMA predictive")
```

Compare to the 'truth':

```
> pbdens=dpairbeta.grid(par=PBpar, displ=FALSE, equi=T, npoints=60,eps=1e-3)
> nldens=dnestlog.grid(par=NLpar, displ=FALSE,  equi=T, npoints=60,eps=1e-3)
> dev.new()
> dgridplot(0.5*(pbdens)+0.5*(nldens), equi=T, npoints=60, eps=1e-3,
+        main="Truth")
```

To obtain a quantitative assessment of the gain represented by the BMA approach, you may compare the Kullback-Leibler (KL) divergence and the L^2 distance between the true density and each estimate:

```
> ## choose a grid size
> npoints=100
> eps=1e-3
> ##
> ## compute the true density on this grid
> TRUEdens=0.5*dnestlog.grid(par=NLpar, equi=F, npoints=npoints,eps=eps,
+                            displ=FALSE )+
+   0.5*dpairbeta.grid(par=PBpar,equi=F,npoints=npoints,eps=eps,displ=FALSE)
```

```
> ##
> ## check that the grid is fine enough
> scores3D(true.dens=TRUEdens, est.dens=TRUEdens, npoints=npoints,
+          eps=eps)
> ##
> ## compute the posterior predictive:
> #### in the PB model:
> rectPBpred=posterior.predictive.pb(post.sample=PBpost,from=NULL, to=NULL,
+   lag=40,npoints=npoints,eps=eps,equi=FALSE, displ=FALSE)
> #### in the NL model:
> rectNLpred=posterior.predictive.nl(post.sample=NLpost,from=NULL, to=NULL,
+   lag=40,npoints=npoints,eps=eps,equi=FALSE, displ=FALSE)
> ##
> ## Finally, compare the performance scores:
> #### PB scores
> scores3D(true.dens=TRUEdens, est.dens=rectPBpred, npoints=npoints,
+          eps=eps)
> #### NL scores
> scores3D(true.dens=TRUEdens, est.dens=rectNLpred, npoints=npoints,
+          eps=eps)
> #### BMA scores
> scores3D(true.dens = TRUEdens,
+          est.dens = 0.31*rectPBpred+
+             0.69*rectNLpred,
+          npoints=npoints,
+          eps=eps)
```

The BMA predictive spectral measure overcomes each individual model's predictive, both for the L^2 score and the logarithmic score.

2.2 Tri-variate Leeds dataset

(see [6] or [3])

Let us consider the tri-variate air pollutant concentrations data provided with the package.

Again, estimate the posterior weights:

```
>   pWeiLeeds=posteriorWeights (dat=Leeds,
+                   HparList=list( pb.Hpar, nl.Hpar ),
+                   lklList=list(dpairbeta , dnestlog ),
+                   priorList=list(prior.pb, prior.nl ),
+                   priorweights=c(0.5,0.5),
+                   Nsim=50e+3,
+                   Nsim.min=10e+3, precision=0.1,
+                   displ=TRUE)
> pWeiLeeds
```

The NL model obtains an overwhelming posterior weight. The BMA framework thus selects a single model. It is unnecessary to compute the parameter posterior

distribution in the PB model, since averaging estimates will be the same as selecting the NL estimate.

This example is not an exceptional case: with increasing sample size, the BMA is bound to become a selecting tool. In the asymptotic limit, the model which minimizes the Kullback-Leibler distance to the 'truth' will be chosen. See *e.g.* [1, 5].

References

[1] R.H. Berk. Limiting behavior of posterior distributions when the model is incorrect. *The Annals of Mathematical Statistics*, 37(1):51–58, 1966.

[2] SG Coles and JA Tawn. Modeling extreme multivariate events. *JR Statist. Soc. B*, 53:377–392, 1991.

[3] D. Cooley, R.A. Davis, and P. Naveau. The pairwise beta distribution: A flexible parametric multivariate model for extremes. *Journal of Multivariate Analysis*, 101(9):2103–2117, 2010.

[4] E.J. Gumbel. Distributions des valeurs extrêmes en plusieurs dimensions. *Publ. Inst. Statist. Univ. Paris*, 9:171–173, 1960.

[5] B.J.K. Kleijn and A.W. van der Vaart. Misspecification in infinite-dimensional bayesian statistics. *The Annals of Statistics*, 34(2):837–877, 2006.

[6] A. Sabourin, P. Naveau, and A.-L. Fougères. Bayesian model averaging for multivariate extremes. *Extremes*, pages 1–26, 2013.

[7] J.A. Tawn. Modelling multivariate extreme value distributions. *Biometrika*, 77(2):245, 1990.