Extreme Value Mixture Modelling: `evmix` Package and Simulation Study

8th Conference on Extreme Value Analysis (EVA2013)

Carl Scarrott & Yang Hu

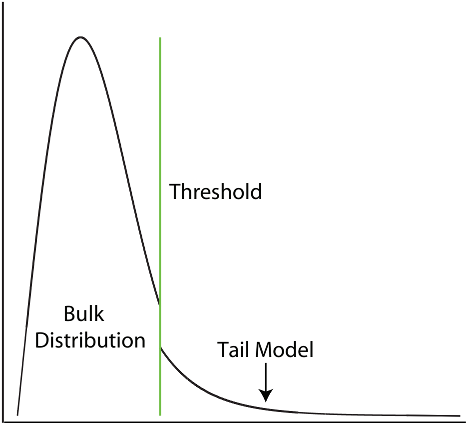
University of Canterbury

July 17, 2013

Talk Outline

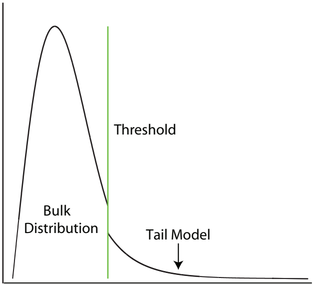
- Introduction to Extreme Value Mixture Models
- Generalised Framework
- New `evmix` package on CRAN
- Simulation Study
- Some Advice

Why Use Extreme Value Mixture Models?



- Provide automated and objective “threshold” estimation
- Or avoid threshold choice together
- Allow for threshold uncertainty to be taken into account
- **Key issue: sensitivity of tail fit to that of bulk**

Some Terminology



- Tail model typically generalised Pareto distribution (GPD) or Poisson point process threshold excess model
- Bulk model has many forms, "loosely" categorised:
  - **parametric**: normal, Weibull, gamma, log-normal, beta
  - **semi-parametric**: mixtures of gamma, normal, log-normal
  - **nonparametric**: mixture of uniforms, kernel density estimation, smoothing splines

Tail Fraction When Using GPD

- When using $X|X > u \sim GPD(\sigma_u, \xi)$ for tail modelling of exceedances:

$$P(X > x|X > u) = \begin{cases} 1 - \left[1 + \xi\left(\frac{x-u}{\sigma_u}\right)\right]_+^{-1/\xi} & \xi \neq 0, \\ 1 - \exp\left[-\left(\frac{x-u}{\sigma_u}\right)\right]_+ & \xi = 0. \end{cases}$$

- Another implicit parameter, the tail fraction above the threshold $\phi_u = P(X > u)$:

$$P(X > x) = \phi_u P(X > x|X > u)$$

- Usually estimated using sample proportion which is the maximum likelihood estimate
- Classic GPD tail modelling approach

Common Mixture Model Structure

- Common mixture model specification for cumulative distribution function:

$$F(x) = \begin{cases} H(x) & x \leq u, \\ H(u) + (1 - H(u))G(x) & x > u. \end{cases}$$

- $H(x)$ is bulk cdf and $G(x)$ is the GPD or other tail model for exceedances
- Tail fraction is specified by the bulk model (parameters):
  - $\phi_u = 1 - H(u)$
- Terminology: **bulk model parameterised approach**

- Essentially, borrowing information from bulk where you have more data
- Question: sensitivity to bulk?

General Mixture Model Structure

- Some mixture models use a more general specification:

$$F(x) = \begin{cases} H(x)\dfrac{1-\phi_u}{H(u)} & x \le u, \\ (1-\phi_u) + \phi_u G(x) & x > u. \end{cases}$$

- Extra explicit parameter ϕ_u for tail fraction
- Rescaling of bulk $\frac{1-\phi_u}{H(u)}$ ensures density integrates to unity
- Closer to classical GPD tail modelling approach
- Includes bulk model parameterised approach as special case
- Terminology: **parameterised tail fraction approach**

- Question: performance loss as not borrowing from bulk?
- Question: more robust to bulk misspecification?
- Question: overfitting due to extra parameter?

Further Niceties

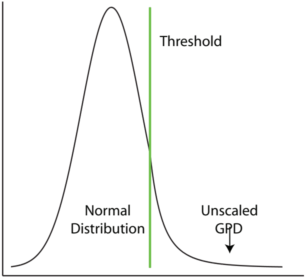
- Notice that base mixture model has no requirement of density to be continuous at threshold
- Note: cdf is continuous, just density is not
- Usually physically sensible to have continuous density
- Some mixture models have parameter constraints to make density:
  1. continuous at threshold, and
  2. continuous in first derivative at threshold
- Continuity is easy for GPD, as at threshold density is just $1/\sigma_u$
- So the GPD scale can be specified as $\sigma_u = \frac{1}{h(u)}$
- Continuity in first derivative is less straightforward, depending on complexity of bulk model

- Question: does performance improve using such physically sensible constraints?

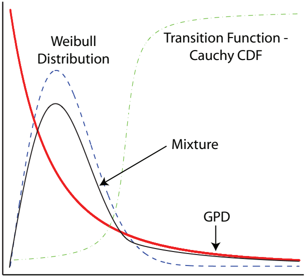
Mixture Models Outside This Framework?

- Couple of mixture models do not fall into this framework:

Hybrid Pareto



Dynamically Weighted Mixture



- Hybrid Pareto has no scaling of the GPD by the tail fraction
- GPD is simply spliced at threshold and then entire density is renormalised to unity
- Different to classical GPD tail approach
- Dynamically weighted mixture has GPD defined over entire range support, so have to be careful as GPD can be influenced by lower tail fit

`evmix` Package in `R`

- General goal:
  - Suite of tools for extreme value threshold estimation and uncertainty quantification
- Named after `evd` package as similar syntax for basic GPD and threshold diagnostic plots
- First (beta) release has:
  - most extreme value mixture models in the current literature
  - model fit diagnostic plots for all of them
  - additional functionality added to threshold diagnostic plots
- Available on CRAN for download
- Beta release... so functionality may change in future releases
- Any feedback and bug reports welcome!

evmix Syntax

- Function follow usual naming conventions, e.g. for gamma bulk with GPD for tail:
  - `dgammagpd` - density function
  - `pgammagpd` - cumulative distribution function
  - `qammagpd` - quantile function
  - `rgammagpd` - random number generation
  - `lgammagpd` - (log-)likelihood function
  - `nlgammagpd` - negative log-likelihood function
  - `fgammagpd` - maximum likelihood estimation fitting
- Fitting function provides sensible initial values for parameters for numerical optimisation routines
- `evmix.diag` function provides usual four model fit diagnostics for all mixture models:
  - return level plot;
  - QQ and PP plots
  - density plots
- `tcplot` and `mrlplot` provide threshold stability plots and mean residual life plots respectively

Implemented Parametric Models

- Implemented **parametric** mixture models are:
  - Normal + GPD (Behrens, et al 2004)*
  - Gamma + GPD (Behrens, et al 2004)*
  - Weibull + GPD mixture distribution (Behrens, et al 2004)*
  - Lognormal + GPD (Solari and Losada, 2004)*
  - Beta GPD mixture distribution (MacDonald, 2012)
  - Two tailed GPD-Normal-GPD (Zhao, et al 2010, Mendes and Lopes, 2004)*
  - Dynamically weighted mixture distribution (Frigessi, 2003)
  - Hybrid Pareto distribution (Carreau and Bengio, 2009)*
- * means a single continuity constraint version also available

Implemented Semi and Nonparametric Models

- Implemented **semi-parametric** mixture models are:
  - Mixture of gammas + GPD (Nascimento, 2011)
- Implemented **nonparametric** mixture models are:
  - Kernel density estimator + GPD (MacDonald, et al. 2011) *
  - Two tailed kernel (MacDonald, et al. 2011) *
  - Boundary corrected kernel density estimator + GPD (MacDonald, et al 2013)
- Many different boundary correction kernel density methods implemented in the latter (renormalisation, reflection, generalised jackknifing, log transform, beta, gamma, gaussian copulas)

Example Usage 1

- Here is an example of fitting the variants of the normal bulk with GPD tail:

```
set.seed(0)
x = rnorm(1000)

# Assume bulk model parameterisation by default
fit = fnormgpd(x)

xx = seq(-5, 5, 0.01)
hist(x, breaks = 100, freq = FALSE)
lines(xx, dnormgpd(xx, fit$nmean, fit$nsd, fit$u, fit$sigma, fit$xi),
  col="blue",lwd=2)
abline(v = fit$u, col="blue")

# Change to parameterised approach
fit2 = fnormgpd(x, phiu=FALSE)

lines(xx, dnormgpd(xx, fit2$nmean, fit2$nsd, fit2$u, fit2$sigma, fit2$xi, fit2$phiu),
  col="red",lwd=2)
abline(v = fit2$u, col="red")

# Add continuity constraint
fit3 = fnormgpdcon(x)

lines(xx, dnormgpdcon(xx, fit3$nmean, fit3$nsd, fit3$u, fit3$xi, fit3$phiu),
  col="green",lwd=2)
abline(v = fit3$u, col="green")
```

Example Usage 2

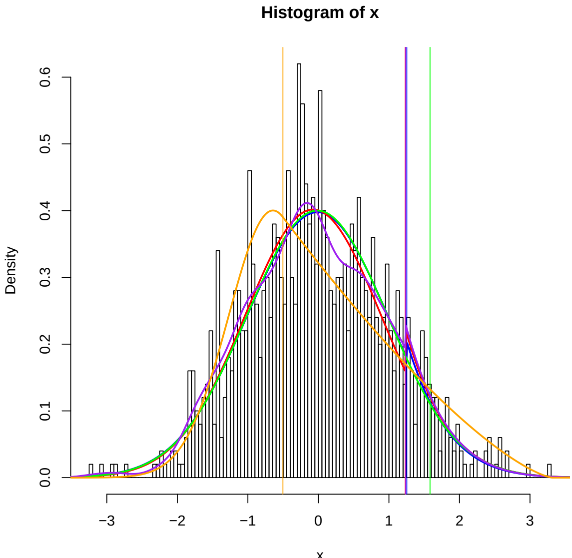
- Nonparametric KDE's use cross-validation likelihood so much slower, go and grab a coffee!:

```
# Nonparametric bulk fit
fit4 = fkdengpd(x)

lines(xx, dkdengpd(xx, x, fit4$lambda, fit4$u, fit4$sigmau, fit4$xi, fit4$phiu),
  col="purple",lwd=2)
abline(v = fit4$u, col="purple")

# Hybrid Pareto
fit5 = fhpd(x)

lines(xx, dhpd(xx, fit5$nmean, fit5$nsd, fit5$xi), col="orange",lwd=2)
abline(v = fit5$u, col="orange")
```

- Future updates will implement K-fold cross-validation and to permit fixed bandwidth estimated a-priori, to speed up computations for nonparametric KDE fitting

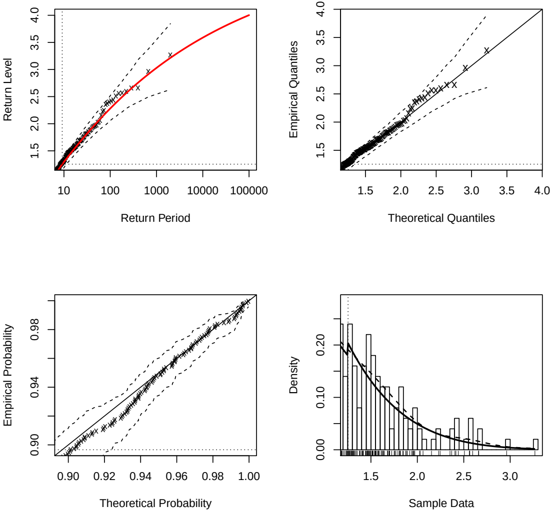
Example Usage 3

- Notice hybrid Pareto suffers as it has no tail fraction scaling of GPD



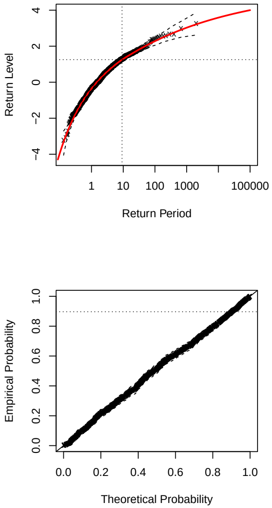
Model Fit Diagnostics

```
# Usual model diagnostics default to focus on upper tail
evmix.diag(fit)
```



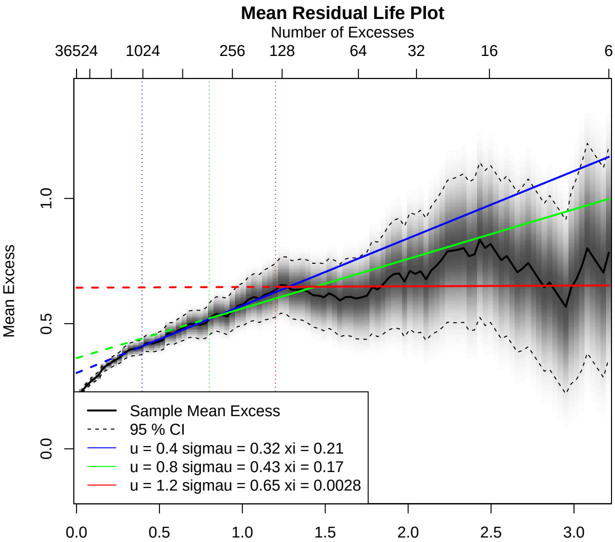
Model Fit Diagnostics

```
# Can see entire fit
evmix.diag(fit,upperfocus=FALSE)
```



Threshold Choice Plots

```
data(FtCoPrec,package="extRemes")
```

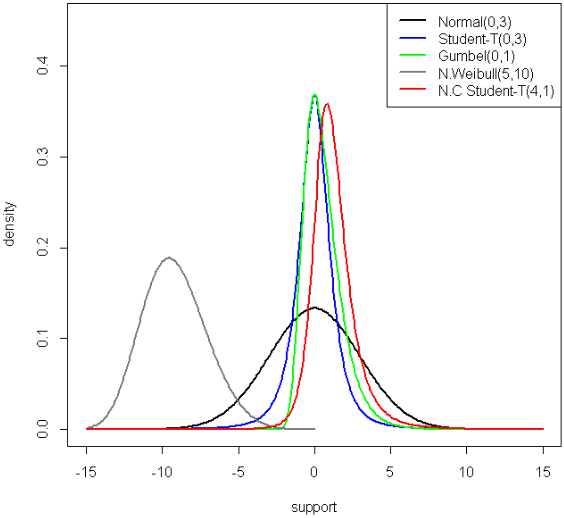


Simulation Study

- Compare all relevant extreme value mixture models depending on range of support:
  - entire reals (one tailed and two-tailed models)
  - positive/non-negative support
- All models fitted using maximum likelihood estimation
- Aim at answering following questions:
  1. In which situations is it best to use the bulk model versus tail fraction parameterised approach?
  2. Do you need the tail fraction?
  3. In which situation is it best to use parametric, semi or nonparametric mixture models?
  4. Does tail quantile estimation gain from continuity constraint?

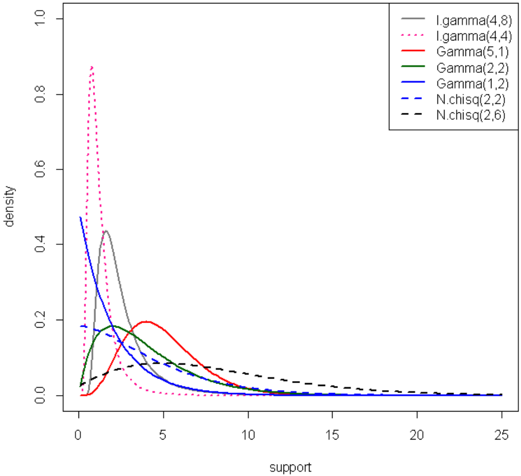
Simulation Setup

- 100 simulations of sample sizes 1000 and 5000
- Variety of different bulk and tail behaviour combinations



Simulation Setup

- Fit using maximum likelihood estimation
- Estimate high quantiles (90, 95, 99, 99.9%)
- Compare performance using RMSE



Population	Model	90%	95%	99%	99.9%
	normal + GPD	0.138	0.190	0.330	0.767
	normal + GPD Cont	**0.128**	**0.163**	**0.282**	**0.728**
Normal($\mu = 0, \sigma = 3$)	hybrid Pareto	1.285	1.441	1.243	0.758
	hybrid Pareto Cont	1.618	1.702	1.383	0.781
	kernel + GPD	0.154	0.179	0.323	0.852
	normal + GPD	**0.292**	0.417	0.766	2.828
	normal + GPD Cont	0.329	0.395	0.695	3.150
$t(v = 3)$	hybrid Pareto	2.432	3.040	3.462	2.770
	hybrid Pareto Cont	2.413	3.032	3.505	**2.657**
	kernel + GPD	0.296	**0.338**	**0.592**	3.009
	normal + GPD	0.207	0.271	0.337	0.793
	normal + GPD Cont	0.121	0.176	**0.273**	**0.773**
Gumbel($\sigma = 1$)	hybrid Pareto	0.222	0.276	0.274	0.867
	hybrid Pareto Cont	0.376	0.434	0.429	0.845
	kernel + GPD	**0.096**	**0.141**	0.294	1.165
	normal + GPD	0.182	0.185	0.222	0.412
	normal + GPD Cont	0.139	0.177	**0.210**	**0.379**
N.Weibull($l = 10, k = 5$)	hybrid Pareto	0.549	0.630	0.503	0.439
	hybrid Pareto Cont	0.952	0.906	0.602	0.441
	kernel + GPD	**0.114**	**0.142**	0.218	0.466
	normal + GPD	0.199	0.305	0.498	2.263
	normal + GPD Cont	0.131	0.198	**0.474**	**1.962**
t($v = 4, \mu = 1$)	hybrid Pareto	0.979	1.202	1.062	2.451
	hybrid Pareto Cont	1.072	1.251	1.000	2.525
	kernel + GPD	**0.106**	**0.145**	0.538	2.915

Population	Model	90%	95%	99%	99.9%
	normal + GPD	0.138	0.190	0.330	0.767
	normal + GPD Cont	**0.128**	**0.163**	**0.282**	**0.728**
Normal($\mu = 0, \sigma = 3$)	hybrid Pareto	1.285	1.441	1.243	0.758
	hybrid Pareto Cont	1.618	1.702	1.383	0.781
	kernel + GPD	0.154	0.179	0.323	0.852
	normal + GPD	**0.292**	0.417	0.766	2.828
	normal + GPD Cont	0.329	0.395	0.695	3.150
$t(v = 3)$	hybrid Pareto	2.432	3.040	3.462	2.770
	hybrid Pareto Cont	2.413	3.032	3.505	**2.657**
	kernel + GPD	0.296	**0.338**	**0.592**	3.009
	normal + GPD	0.207	0.271	0.337	0.793
	normal + GPD Cont	0.121	0.176	**0.273**	**0.773**
Gumbel($\sigma = 1$)	hybrid Pareto	0.222	0.276	0.274	0.867
	hybrid Pareto Cont	0.376	0.434	0.429	0.845
	kernel + GPD	**0.096**	**0.141**	0.294	1.165
	normal + GPD	0.182	0.185	0.222	0.412
	normal + GPD Cont	0.139	0.177	**0.210**	**0.379**
N.Weibull($l = 10, k = 5$)	hybrid Pareto	0.549	0.630	0.503	0.439
	hybrid Pareto Cont	0.952	0.906	0.602	0.441
	kernel + GPD	**0.114**	**0.142**	0.218	0.466
	normal + GPD	0.199	0.305	0.498	2.263
	normal + GPD Cont	0.131	0.198	**0.474**	**1.962**
t($v = 4, \mu = 1$)	hybrid Pareto	0.979	1.202	1.062	2.451
	hybrid Pareto Cont	1.072	1.251	1.000	2.525
	kernel + GPD	**0.106**	**0.145**	0.538	2.915

Population	Model	90%	95%	99%	99.9%
Normal($\mu = 0, \sigma = 3$)	normal + GPD	0.138	0.190	0.330	0.767
	normal + GPD Cont	**0.128**	**0.163**	**0.282**	**0.728**
	kernel + GPD	0.154	0.179	0.323	0.852
$t(v = 3)$	normal + GPD	**0.292**	0.417	0.766	**2.828**
	normal + GPD Cont	0.329	0.395	0.695	3.150
	kernel + GPD	0.296	**0.338**	**0.592**	3.009
Gumbel($\sigma = 1$)	normal + GPD	0.207	0.271	0.337	0.793
	normal + GPD Cont	0.121	0.176	**0.273**	**0.773**
	kernel + GPD	**0.096**	**0.141**	0.294	1.165
N.Weibull($l = 10, k = 5$)	normal + GPD	0.182	0.185	0.222	0.412
	normal + GPD Cont	0.139	0.177	**0.210**	**0.379**
	kernel + GPD	**0.114**	**0.142**	0.218	0.466
t($v = 4, \mu = 1$)	normal + GPD	0.199	0.305	0.498	2.263
	normal + GPD Cont	0.131	0.198	**0.474**	**1.962**
	kernel + GPD	**0.106**	**0.145**	0.538	2.915

- If correct bulk model (i.e. normal) then better to use continuity constraint
- In general, results are mixed but overall small gain
- If correct bulk model then better than kernel, but not by much
- In general, kernel best for low quantiles and broadly similar performance for highest quantiles

Bulk Model Parameterisation	Model	90%	95%	99%	99.9%
Normal($\mu = 0, \sigma = 3$)	normal + GPD	0.138	0.190	0.330	0.767
	normal + GPD Cont	**0.128**	**0.163**	**0.282**	**0.728**
	kernel + GPD	0.154	0.179	0.323	0.852
$t(v = 3)$	normal + GPD	**0.292**	0.417	0.766	**2.828**
	normal + GPD Cont	0.329	0.395	0.695	3.150
	kernel + GPD	0.296	**0.338**	**0.592**	3.009
Gumbel($\sigma = 1$)	normal + GPD	0.207	0.271	0.337	0.793
	normal + GPD Cont	0.121	0.176	**0.273**	**0.773**
	kernel + GPD	**0.096**	**0.141**	0.294	1.165
Parameterised Tail Fraction	Model	90%	95%	99%	99.9%
Normal($\mu = 0, \sigma = 3$)	normal + GPD	0.176	0.186	**0.299**	0.753
	normal + GPD Cont	0.162	0.210	0.325	0.739
	kernel + GPD	**0.155**	**0.168**	0.304	**0.710**
$t(v = 3)$	normal + GPD	0.143	0.207	0.507	2.586
	normal + GPD Cont	0.114	0.203	**0.411**	**2.432**
	kernel + GPD	**0.089**	**0.140**	0.529	2.539
Gumbel($\sigma = 1$)	normal + GPD	0.112	0.133	0.264	**0.778**
	normal + GPD Cont	0.134	0.208	0.379	1.050
	kernel + GPD	**0.077**	**0.118**	**0.261**	0.863

- If correct bulk model then better to use bulk model parameterisation
- In general, better to use parameterised tail fraction which is consistent with classical GPD tail modelling approach

Bulk Model Parameterisation	Model	90%	95%	99%	99.9%
Normal($\mu = 0, \sigma = 3$)	normal + GPD	0.138	0.190	0.330	0.767
	normal + GPD Cont	**0.128**	**0.163**	**0.282**	**0.728**
	kernel + GPD	0.154	0.179	0.323	0.852
$t(v = 3)$	normal + GPD	**0.292**	0.417	0.766	**2.828**
	normal + GPD Cont	0.329	0.395	0.695	3.150
	kernel + GPD	0.296	**0.338**	**0.592**	3.009
Gumbel($\sigma = 1$)	normal + GPD	0.207	0.271	0.337	0.793
	normal + GPD Cont	0.121	0.176	**0.273**	**0.773**
	kernel + GPD	**0.096**	**0.141**	0.294	1.165
Parameterised Tail Fraction	Model	90%	95%	99%	99.9%
Normal($\mu = 0, \sigma = 3$)	normal + GPD	0.176	0.186	**0.299**	0.753
	normal + GPD Cont	0.162	0.210	0.325	0.739
	kernel + GPD	**0.155**	**0.168**	0.304	**0.710**
$t(v = 3)$	normal + GPD	0.143	0.207	0.507	2.586
	normal + GPD Cont	0.114	0.203	**0.411**	**2.432**
	kernel + GPD	**0.089**	**0.140**	0.529	2.539
Gumbel($\sigma = 1$)	normal + GPD	0.112	0.133	0.264	**0.778**
	normal + GPD Cont	0.134	0.208	0.379	1.050
	kernel + GPD	**0.077**	**0.118**	**0.261**	0.863

- If correct bulk model then better to use bulk model parameterisation
- In general, better to use parameterised tail fraction which is consistent with classical GPD tail modelling approach

Population	Model	90%	95%	99%	99.9%
	gamma + GPD	0.199	0.258	0.548	1.631
	gamma + GPD Cont	0.171	0.252	**0.472**	1.542
	Weibull + GPD	0.211	0.251	0.481	1.544
Gamma(1, 2)	Weibull + GPD Cont	0.145	0.236	0.500	1.656
	Mix. gamma + GPD	0.188	0.268	0.548	1.545
	Dynamic. Weigh. Mix.	0.166	**0.235**	0.561	1.719
	Bound. Correct. KDE + GPD	**0.163**	0.246	0.558	**1.435**
	gamma + GPD	0.180	**0.190**	**0.342**	0.974
	gamma + GPD Cont	**0.155**	0.222	0.397	1.076
	Weibull + GPD	0.182	0.222	0.352	1.080
Gamma(5, 1)	Weibull + GPD Cont.	0.168	0.246	0.434	**0.926**
	Mix. gamma + GPD	0.166	0.203	0.412	1.061
	Dynamic. Weigh. Mix.	0.314	0.443	0.549	1.089
	Bound. Correct. KDE + GPD	0.162	0.217	0.356	1.055
	gamma + GPD	0.171	0.232	**0.658**	3.229
	gamma + GPD Cont	0.183	0.279	0.723	**3.025**
	Weibull + GPD	0.180	0.253	0.773	3.698
Inverse gamma(4, 8)	Weibull + GPD Cont	0.194	0.298	0.673	3.193
	Mix. gamma + GPD	0.171	0.293	0.752	3.520
	Dynamic. Weigh. Mix.	0.337	0.396	0.850	3.858
	Bound. Correct. KDE + GPD	**0.158**	**0.249**	0.764	3.449

Advice for Practitioners and Future Mixture Models

1. If the bulk model is correct, then it will perform well and you should use the bulk model approach for the tail fraction. A smaller advantage is also achieved if the density is constrained to be continuous at the threshold.
2. If bulk model is mis-specified (i.e. unknown population), then generally better to use parameterised tail fraction as this seems to robustify the tail fit to that in bulk. BUT(!), there is little to be gained by the continuity constraint at the threshold and in some situations it reduces this robustness so should be avoided.
3. The hybrid Pareto model frequently performs poorly but this is easily remedied by including the tail fraction.
4. If the bulk model is correctly specified, then the parametric mixture models are easy to understand and quick to fit so are preferred. However, in more usual situation of unknown population distribution, the nonparametric mixture models perform consistently well for low and high quantiles.
5. Note: limitation on results so far - no penalty for complexity

References and Website

Review paper:

Scarrott and MacDonald (2012). A review of extreme value threshold estimation and uncertainty quantification. REVSTAT Statistical Journal 10(1), 33-60.
(all references in here)

Package: `evmix` available on CRAN (all feedback appreciated)

Website:
`http://www.math.canterbury.ac.nz/~c.scarrott/evmix`

Yang Hu's thesis with all simulation results on website soon!

Thanks for your attention...