

Introductory statistical ideas for IOGP: An “in other words” summary of the workshop

Philip Jonathan

September 2026

1 Tails of univariate distributions

- The distribution of the maximum of n copies of **any** random variable X with a “nice” distribution converges as n increases to the **generalised extreme value** (GEV) distribution. The properties of this distribution are determined by its **shape** parameter ξ .
- The conditional tail of the distribution of X above a high threshold u converges as u increases to the **generalised Pareto** (GP) distribution. The properties of this distribution are determined by the (same common) **shape** parameter ξ .
- If block maxima are GEV, then you can prove that conditional threshold exceedances are GP (with the same ξ).
- If conditional threshold exceedances are GP, and the rate of occurrence of threshold exceedances is Poisson, then the block maxima are GEV (with the same ξ).
- This asymptotic theory works for “nice” distributions like those observed in ocean engineering and environmental science.
- It makes sense we should use this theory when we have enough data, and are confident about being approximately asymptotic.
- There is still a place for the simplest possible models (like “engineering Weibull”), especially when sample is small, or of poor quality; but results (e.g. return value estimates) are likely to be biased.
- We aim for a thoughtful marriage of our engineering **and** statistical knowledge.
- **All models are wrong, some models are useful!**

2 Diagnosing GP fit to a sample of peaks over threshold (POT)

- Examine the variation of $\hat{\xi}$ as a function of u . Theory tells us that $\hat{\xi}$ should not change, if GP is a good model for the data.
- Use bootstrapping to quantify the uncertainty of $\hat{\xi}$ for every u ; in the “Goldilocks region” or “sweet spot” we should be able to draw a horizontal line without crossing the error bars.

3 Key (frequentist) model fitting tools

- Maximum likelihood estimation (MLE): to estimate model parameters.
 - Every observation has a probability density with respect to the proposed statistical model; this probability density can be thought of as a function of the model parameters.
 - Multiply all these “observation densities” together to get the **sample likelihood** $\mathcal{L}$.
 - $\mathcal{L}$ is also a function of the model parameters; so we can maximise $\mathcal{L}$ over the parameters to get the **maximum likelihood estimates** of the parameters.
 - MLE is used everywhere; and has well-understood asymptotic behaviour (i.e. for large samples).
 - There are other approaches that work well under specific circumstances; such as for very small sample sizes.
- Bootstrapping: for **uncertainty quantification** (UQ); i.e. to estimate error bars.
 - Resample the original sample with replacement, and do MLE on the new “pseudo-sample”
 - Repeat for lots of “pseudo-samples” to accumulate distributions of parameter estimates, and hence to estimate error bars for parameter estimates.
- Cross-validation: to estimate **out-of-sample predictive** performance.
 - Split the data into blocks (or “folds”); use all blocks **but one** for model fitting, and use the remaining block as an **independent test set**.
 - Accumulate predictive performance over all “left out” blocks.
- Cross-validation: to estimate optimal choice of model hyper-parameters (i.e. modelling “fiddle factors” that you can’t estimate using MLE).
 - Run the full cross-validation procedure for different choices of hyper-parameter.
 - Choose the hyper-parameter value which maximises out-of-sample predictive performance.
- Model averaging (or ensemble models).
 - If there is no engineering or statistical evidence to suggest one model is better than any other, then **use them all** for inference.
 - We can also estimate different “weights” for ensemble components based on their anticipated performance; Bayesian inference good for this.
- Penalised maximum likelihood estimation.
 - Add a multiplicative penalty factor to the sample likelihood: $\mathcal{L}^* = \mathcal{L} \times \exp(-C)$, where both $\mathcal{L}$ and C are functions of the model parameters.
 - This is a rigorous way of including “prior knowledge” or “constraints” on parameters.
 - Maximise $\mathcal{L}^*$ instead of $\mathcal{L}$ to include the “information about the constraints” in the inference.
 - Penalised MLE has a natural **Bayesian** interpretation.

4 Covariate models

- What happens if we ignore covariate effects? We don't know (and never will in general!)
 - Our toy case study suggested we could have return value estimates biased high, biased low, or even not biased!
 - We should not leave metocean design to luck.
- How do we handle covariates in POT? We let the GP tail change smoothly as a function of covariate.
 - The GP shape and scale parameters are smooth functions of covariate.
 - We describe parameter variation on the covariate domain using a spline.
 - We use penalised MLE, with constraint C imposing smoothness on the spline.
 - We **optimise** spline smoothness to obtain best **out-of-sample predictive** performance.
- How do we build a full non-stationary extreme value (EV) model for metocean (say starting from storm peak events, assumed independent given covariates)?
 - EV threshold is estimated using a non-stationary Gamma model.
 - Rate of threshold exceedance is estimated using non-stationary Poisson model.
 - Size of threshold exceedance is estimated using a non-stationary GP model.
 - We fit all models using a common inference “compute engine”; all we need to do is swap the input data and the “density” used for the different models.
 - We quantify uncertainty by incorporating multiple plausible thresholds and multiple bootstrap resamples.
- We have fitted covariate models in 4 dimensions on occasion, but one or two dimensions (e.g. directional-seasonal H_S) is commonplace.
- Case studies have shown that the details of the spline covariate representation are not too critical; so we should use the approach which is computationally simplest!

5 Multivariate extremes

- We want to understand the tail region of a multivariate distribution; simplest example: bivariate $P(X_1 > x_1, X_2 > x_1)$, when at least one of x_1 or x_2 is big.
- Our toy study of the bivariate Gaussian and the bivariate logistic showed that these distributions have very different tail behaviour.
 - $P(X_1 > x, X_2 > x)/P(X_1 > x) \rightarrow 0$ with increasing x for Gaussian, but not for logistic!!
 - $\log P(X_1 > x, X_2 > x)/\log P(X_1 > x) \rightarrow 1$ for logistic, but not for Gaussian!
- In other words, the joint tail decays at the same rate as the marginal tail for logistic, but much more quickly for Gaussian!
- The ratios are one way to understand **asymptotic dependence** (AD; slow decay, e.g. logistic) and **asymptotic independence** (AI; fast decay, e.g. Gaussian)
- If we assume AI when we have AD, we will underestimate associated values; we will overestimate in the reverse case.

- We use the ratios above to test from the sample if we have AD or AI
- Remember the financial crash and **mea copula!**
- And remember how we can see AD and AI by plotting samples on different marginal scales (e.g. Frechet, Laplace, exponential).
- Historically, models emerging from statistics theory and literature assumed AD; then people realised that AI was important. We need a simple model that does both for met-ocean.
- Conditional extremes model.
 - Transform data for each variable to standard Laplace scale.
 - Fit the Heffernan-Tawn (HT) regression model.
 - Simulate under the HT model, and transform the simulated data back to physical scale to estimate return values and associated values.
- Model fitting again uses penalised MLE; multiple threshold ensemble and bootstrapping for UQ, cross-validation for assessment of predictive performance; we can also estimate non-stationary conditional extremes models, with cross-validation for optimal selection of spline smoothness constraint.
- A typical engineering approach (Weibull-log-normal for H_S-T_P) does not have the same theoretical “underpinning” as conditional extremes, but it may still be a reasonable approach when sample size is small.

6 covXtreme

- Tools being used and developed for met-ocean applications (SAM, Gibson, DNV, met.no, Shell, ...) all use the same basic approach.
 - Marginal and conditional extremes models, with spline representations for parameter variation on covariate domain.
 - Either penalised MLE / bootstrapping / cross-validation, or Bayesian “compute engines”.
- covXtreme provides an entry level tool which incorporates all the key features of the penalised MLE / bootstrapping / cross-validation engine, with some assumptions (like constant ξ , piecewise constant covariate dependence for other parameters) for computational efficiency
 - Gamma EV threshold
 - Poisson threshold exceedance
 - GP size of threshold exceedance
 - Multiple (marginal and dependence) threshold ensembles and bootstrapping for UQ
 - Cross-validation for optimal hyper-parameter selection
- Good teaching, self-guided learning tool; but also good benchmark for more “complex” approaches
- Developed by stats group in Shell; open-source; code, data and case studies on github. Looks like there may be a PYTHON version soon!