Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação

Nicole Santos e Aguiar

A multitask learning approach to automatic threshold selection in Pareto distributions

Uma abordagem multitarefa para seleção automática de limiar em distribuções de Pareto

CAMPINAS
2019

Nicole Santos e Aguiar

A multitask learning approach to automatic threshold selection in Pareto distributions

Uma abordagem multitarefa para seleção automática de limiar em distribuções de Pareto

Dissertation presented to the School of Electrical and Computer Engineering of the University of Campinas in partial fulfillment of the requirements for the degree of Master in Electrical Engineering, in the area of Computer Engineering.

Dissertação apresentada à Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestra em Engenharia Elétrica, na área de Engenharia de Computação.

Orientador: Prof. Dr. Fernando José Von Zuben

Este exemplar corresponde à versão final da dissertação da aluna Nicole Santos e Aguiar, orientada pelo Prof. Dr. Fernando José Von Zuben.

Campinas
2019

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Luciana Pietrosanto Milla - CRB 8/8129

Aguiar, Nicole Santos e, 1994-
Ag93m A multitask learning approach to automatic threshold selection in Pareto distributions / Nicole Santos e Aguiar. – Campinas, SP : [s.n.], 2019.

Orientador: Fernando José Von Zuben.
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação.

1. Teoria dos valores extremos. 2. Multitarefa (Computação). 3. Aprendizado de máquina. I. Von Zuben, Fernando José, 1968-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Elétrica e de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Uma abordagem multitarefa para seleção automática de limiar em distribuições de Pareto
Palavras-chave em inglês:
Theory of extreme values
Multitasking (Computing)
Machine learning
Área de concentração: Engenharia de Computação
Titulação: Mestra em Engenharia Elétrica
Banca examinadora:
Fernando José Von Zuben [Orientador]
Mateus Giesbrecht
Guilherme Palermo Coelho
Data de defesa: 11-12-2019
Programa de Pós-Graduação: Engenharia Elétrica

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0001-5393-2985
- Currículo Lattes do autor: http://lattes.cnpq.br/6450727973606482

COMISSÃO JULGADORA - DISSERTAÇÃO DE MESTRADO

Candidata: Nicole Santos e Aguiar RA: 147538

Data da Defesa: 11 de dezembro de 2019

Título da Tese: "A multitask learning approach to automatic threshold selection in Pareto distributions".

Prof. Dr. Fernando J. Von Zuben (Presidente, FEEC/UNICAMP)
Prof. Dr. Guilherme Palermo Coelho (FT/Unicamp)
Prof. Dr. Mateus Giesbrecht (FEEC/UNICAMP)

A ata de defesa, com as respectivas assinaturas dos membros da Comissão Julgadora, encontra-se no SIGA (Sistema de Fluxo de Dissertação/Tese) e na Secretaria de Pós-Graduação da Faculdade de Engenharia Elétrica e de Computação.

To my parents and my friends.

Acknowledgements

I would like to extend my heartfelt gratitude and appreciation to the people that somehow composed this path with me, especially

to my parents, for always being my safe harbor and, regardless of any difficulty, for always being by my side and support my decisions;

to my advisor, Prof. Fernando J. Von Zuben, for the patience, guidance and continuous encouragement throughout this research;

to my colleagues of LBiC and Labore, for the multiple technical discussions, cooperation and laughs shared, especially to Marcos, that kindly dedicated his time to contribute to the early stages of foundation and also to set the experimental design. The journey that culminated with the Master degree would be so much harder without you all;

to my friends of BFFFA&B, for being present in another stage of my life and sharing happiness and growth moments.

This study was financed in part by the "Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES)" - Finance Code 001 and the São Paulo Research Foundation (FAPESP), grant #2018/09887-1.

"Chaos is order yet undeciphered."
José Saramago.

Abstract

The main objective of this work is to present a multitask, efficient and automatic approach to estimate thresholds for a generalized Pareto distribution, aiming at high-performance prediction of extremes in multiple precipitation time series. Based on Extreme Value Theory, the only information used to model the heavy tail distribution by maximum likelihood estimation is given by the samples of the time series exceeding a user-defined threshold. This approach suffers from two fundamental drawbacks: (1) the subjectivity of the threshold definition, even when resorting to some graphical guidance, (2) the inherent sparse nature of the above-threshold samples, which, by definition, belong to the tail of the distribution. The proposal presented here for multitask learning automatically creates a hierarchical relationship among the prediction tasks and uses a nested cross-validation to automatize the choice of the optimal thresholds. Given the obtained hierarchical relationship among the prediction tasks, the multitask learning explores data from multiple related prediction tasks toward a more robust maximum likelihood estimation of the parameters that characterize the generalized Pareto distribution. The proposed methodology was applied to precipitation time series of South America and its performance was compared to a single-task learning method and to the traditional graphical approach, indicating a consistent performance improvement. Another advantage of the approach is the possibility of performing a qualitative interpretation of the obtained hierarchical relationship among the tasks, when associated with the geographical locations of the precipitation time series.

Keywords: Extreme Value Theory; Multitask Learning; Hierarchical Clustering; Automatic Threshold Estimation in Pareto Distributions

Resumo

O principal objetivo deste trabalho é apresentar uma abordagem multitarefa, eficiente e automática para estimar limiares de uma distribuição generalizada de Pareto, visando uma previsão de alto desempenho de extremos em várias séries temporais de precipitação. Com base na teoria dos valores extremos, as únicas informações usadas para modelar uma distribuição de cauda pesada por estimação por máxima verossimilhança são fornecidas pelas amostras da série temporal que excedem um limiar definido pelo usuário. Essa abordagem sofre de duas desvantagens fundamentais: (1) a subjetividade na definição do limiar, mesmo quando se recorre a alguma orientação gráfica; (2) a natureza esparsa inerente das amostras acima do limiar, que, por definição, pertecem à cauda da distribuição. A proposta aqui apresentada para aprendizado multitarefa cria automaticamente um relacionamento hierárquico entre as tarefas de predição e usa uma validação cruzada aninhada para automatizar a escolha dos limiares mais indicados. Dada a relação hierárquica obtida entre as tarefas de predição, o aprendizado multitarefa explora os dados de várias tarefas de predição relacionadas para uma estimativa de máxima verossimilhança dos parâmetros que caracterizam a distribuição generalizada de Pareto mais robusta. A metodologia proposta foi aplicada em séries temporais de precipitação da América do Sul e sua performance foi comparada a um método de aprendizado monotarefa e à abordagem gráfica tradicional, indicando uma melhoria consistente de desempenho. Outra vantagem da abordagem é a possibilidade de realizar uma interpretação qualitativa da relação hierárquica obtida entre as tarefas, quando associada às localizações geográficas das séries temporais de precipitação.

Palavras-chave: Teoria do Valor Extremo; Aprendizado Multitarefa; Clusterização Hierárquica; Seleção Automática de Limiar em Distribuições de Pareto.

List of Figures

List of Tables

List of Symbols

GPD Generalized Pareto Distribution.

GEV Generalized Extreme Value.

EVT Extreme Value Theory.

STL Single-task Learning.

MTL Multitask Learning.

Contents

Chapter 1

Introduction[1]

Extreme climate events, such as intense precipitation, extended droughts, and excessive increase of temperature, are, by definition, rare and potentially of high impact (Seneviratne *et al.* , 2012). Its occurrence tends to produce a wide range of consequences in fields such as economy (Hallegatte *et al.* , 2007) and civil defense (Valverde, 2017). That is why more and more attention has been devoted to investigating the statistical behavior and to properly forecasting these events (Chandra, 2017; Hu & Ayyub, 2019; Iglesias *et al.* , 2015; McGovern *et al.* , 2017), so that the damage and impact that they may cause are prevented/attenuated.

Despite the high potential impact of extreme events, defining them is not an easy task. There are two main methods defining extremes: (1) Generalized Extreme Value (GEV) (Fisher & Tippett, 1928) consists in dividing the observation period into blocks and analyzing only the most extreme value in each block; and (2) Peaks Over Threshold (POT) (Balkema & de Haan, 1974; Pickands, 1975) uses the peaks above a certain threshold to fit a Pareto Distribution (Pareto, 1898). Selecting only the peaks of each block makes GEV simpler, but results in a low number of samples, impairing the generalization performance of resultant fitted models. POT overcomes the disadvantage of GEV since it makes better use of the available data. However, the selection of an appropriate threshold is usually made by visual methods (Coles, 2001), which incorporates errors and uncertainties (Thompson *et al.* , 2009). Additionally, these procedures require prior experience while interpreting threshold choice plots to achieve a satisfactory model fit (Coles & Tawn, 1994).

This subjective and expert-dependent approach to select the threshold motivates the proposition of automatic methods to select the threshold. Thompson *et al.* (2009) presented a method that is based on the difference of the parameter estimates when the threshold is changed, and Fukutome *et al.* (2015) adopted the automation of an existing graphical method to select the threshold that will guide to a proper parameter choice, resorting to a measure of clustering in data. Here we are going to present a more robust and data-intensive proposal based on the joint application of multitask learning and extreme value theory to automatically estimate an appropriate threshold. The use of this technique allows the analysis of multiple time series simultaneously without any previous knowledge on the data or any additional parameter. Furthermore, the structural relationship involving multiple learning tasks can support a qualitative analysis of the joint behavior

[1]The content of this manuscript is essentially based on the content of the submitted paper Aguiar *et al.* (2019).

of the prediction models. The main advantage of a multitask learning algorithm, especially when focused on climate forecasting, is that it allows the information sharing of locations characterized by similar climate events. With a more robust estimation of the threshold, an extreme event that had already happened in some location may influence the prediction of an upcoming extreme in another related location.

In the context of climate forecasting, multitask learning was already applied to predict extreme events with distinguished performance gain when compared to single-task learning. As presented in Chandra (2017), a co-evolutionary algorithm, which incorporates features from distinct models and multitask learning (MTL), is used to predict tropical cyclone wind-intensity. Also related to climate prediction, Gonçalves *et al.* (2015) presented a multitask learning-based method to build high-performance Earth System Models (ESM), based on the joint learning of the structural relationship among the tasks (each point in the Earth surface grid is taken as a distinct prediction task) and of the parameters of the learning models.

In this work, a hierarchical multitask learning approach is proposed to automatically select a threshold in Pareto distributions. The novel proposal automatically conceives a hierarchical structural model involving the prediction of all tasks and uses a nested cross-validation to automatize the choice of the optimal thresholds. This method aims at improving the performance of each task by taking into account the data from other related prediction tasks; the clustering procedure finds similar tasks to construct the hierarchical structure and to suggest which tasks should be held together to improve generalization performance. This method is tested with precipitation time series, aiming at predicting extreme events, taking as contenders the equivalent single-task learning procedure and the traditional graphical approach.

The next chapters are organized as follows. Chapter 2 presents an overview of the theoretical basis of Extreme Value Theory, Cross-Validation, Multitask Learning and Hierarchical Clustering, focusing on key aspects to better understand the proposed method. Chapter 3 introduces the proposed automatic method to threshold selection in Pareto distributions. Chapter 4 describes the experimental setup and discusses the results of two experiments on real-world datasets. Chapter 5 presents final considerations and future perspectives of the research.

Chapter 2

Conceptual background

This chapter will present the main technical concepts used throughout this work. Initially, basic aspects of the Extreme Value Theory and the main approaches usually applied to model the tail of a distribution will be explained in Section 2.1. Next, a machine learning technique, called Cross-Validation, generally applied to evaluate the inherent quality of a learning model is presented in Section 2.2. A literature overview of Multitask Learning, an approach that exploits commonalities across tasks to improve efficiency and prediction accuracy of learning models, is presented in Section 2.3. Finally, the hierarchical clustering model adopted in the proposed methodology to be presented in Chapter 3 is formally described in Section 2.4.

2.1 Extreme Value Theory

The Extreme Value Theory (EVT) was created as a branch of Statistics that aims to estimate extreme events and its impacts in diverse fields, such as financial market, insurance coverage and climate forecasting.

When forecasting extreme events, the focus is in modeling the events of the tail of a distribution, i.e., those that have low probability and high impact. However, tail events data are rare, which justifies the necessity to derive asymptotic properties of the tail, by analogies to the Central Limit Theorem.

In this context, two main approaches are usually applied: Generalized Extreme Value (GEV), which uses the Block Maxima, that consists in dividing the observation period into blocks and analyzes only the most extreme value in each block; and Peaks Over Threshold (POT), which uses the exceedances above a threshold to fit a Pareto Distribution.

The second approach is generally taken as an alternative to the first one, since the main disadvantage of GEV takes place in the presence of few data in a series and/or when series have many missing values.

When the distribution is fitted, it is possible to find the return levels and their periods, which are important to the forecast of extreme events. As illustrated in Figure 1, a high-quality model for a given time series (black) is the starting point to achieve a competent estimation of the subsequent values of this time series (blue). The estimated values, $\hat{x}_t$, are the return levels, that will be further

explained. More details of these approaches are given in Sections 2.1.2 and 2.1.3.

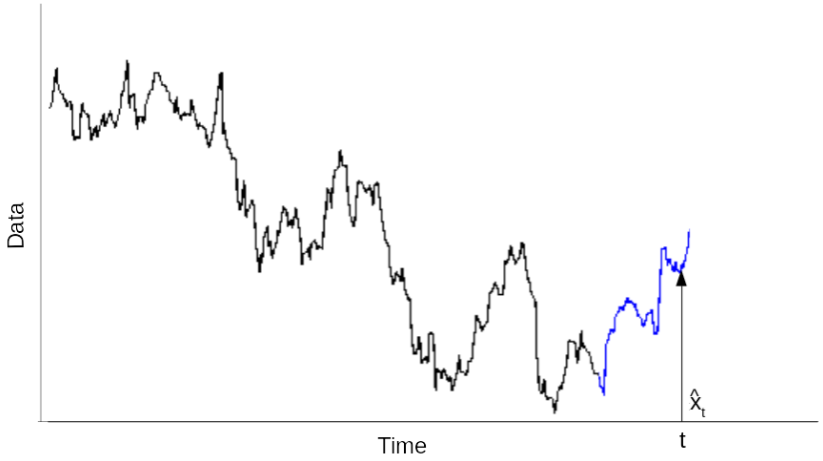


Figure 1: Time series with estimation in blue.

2.1.1 Brief historical context

One of the first investigations devoted to the statistics of extremes was conducted by Bernoulli and in 1709 he answered the following open question: "if n men of equal age die within t years, what is the mean duration of life of the last survivor?". This question can be reduced to "n points are randomly situated in a straight line of size t, what is the largest mean distance to origin?".

Extreme values are necessarily associated with small probabilities. Therefore, the Poisson law must be mentioned, since it considers these probabilities. For 60 years, the Poisson distribution was nothing but a mathematical curiosity, until Von Bortkiewicz (1898) demonstrated its statistical meaning and its relevance to explain natural events. In the next year, R. Von Mises introduced the fundamental notion of the highest characteristic value and indicated its asymptotic relation with the mean of the greatest normal values. In 1925, L. H. C. Tippett calculated the probabilities of the greatest normal values for sizes of different samples until 1000 and the mean normal interval for samples from 2 until 1000.

In 1990, trying to solve an estimation problem of dikes height, after a flood in Netherlands that killed almost two thousand people in 1953, de Haan (1990) formulated a statistical methodology that was the basis for extreme event analysis.

2.1.2 Generalized Extreme Value

Let $X_1, X_2, \ldots, X_n$ be a sequence of independent and identically distributed random variables with a common distribution function F. For $m = 1, 2, \ldots$ and $i = 1, 2, \ldots, k$ the Block Maxima is defined as:

$$M_i = \max_{(i-1)m<j\leq im} X_j \tag{2.1}$$

The $m \times k$ observations are divided into k blocks of size m. The distribution function of the maximum M_k can be described as:

$$P(M_k \leq x) = P(X_1 \leq x, \ldots, X_k \leq x) = F^k(x),\ x \in \mathbb{R},\ k \in \mathbb{N} \tag{2.2}$$

The problem is that such distribution depends on the distribution of the underlying random variables, which is not known in practice. Thus, having access to a proper cumulative asymptotic distribution for a high value of n, would help in modeling extreme events. For a block maxima, the cumulative asymptotic distribution exists and it is described by the Theorem of Fisher & Tippett (1928).

Theorem 1 (Fisher-Tippett theorem, Extreme Value theorem)**.** *Let $X_1, X_2, \ldots, X_n$ be a sequence of independent and identically distributed random variables. If there are norming constants $c_n > 0$, $d_n \in \mathbb{R}$ and a non-degenerated distribution function H such as*

$$c_n^{-1}(M_n - d_n) \xrightarrow{d} H, \tag{2.3}$$

in which $\xrightarrow{d}$ means convergence in distribution, then H belongs to one of the following three distribution functions:

Fréchet:

$$\Phi_\alpha(x) = \begin{cases} 0, & x \leq 0; \\ \exp\{-x^{-\alpha}\}, & x > 0 \end{cases} \quad \alpha > 0. \tag{2.4}$$

Weibull:

$$\Psi_\alpha(x) = \begin{cases} \exp\{-(-x)^{\alpha}\}, & x \leq 0; \\ 1, & x > 0 \end{cases} \quad \alpha > 0. \tag{2.5}$$

Gumbell:

$$\Lambda(x) = \exp\{-e^{-x}\}, \quad x \in \mathbb{R}. \tag{2.6}$$

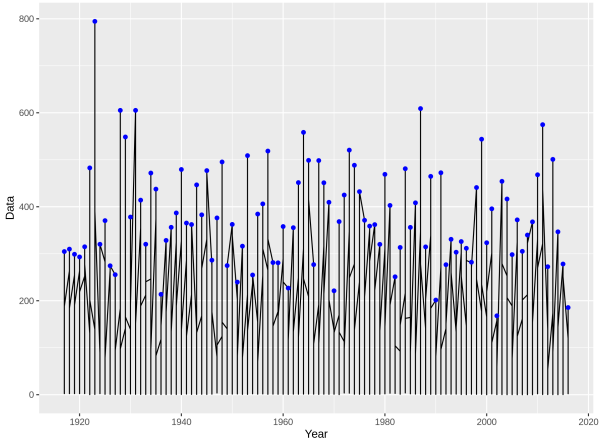
Proof: See Fisher & Tippett (1928).

Consequently, the generalized distribution is described by the following equation:

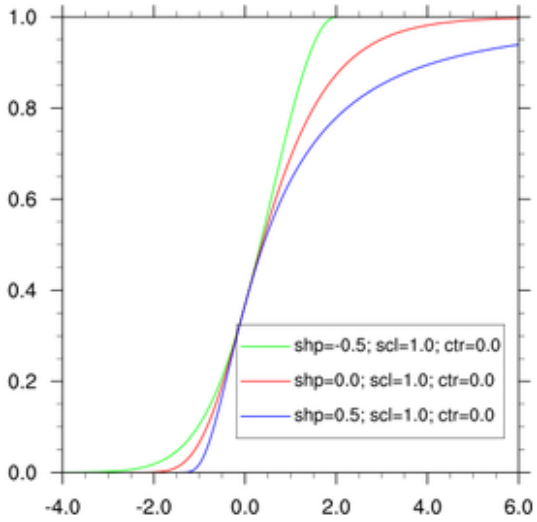
$$H_{\xi,\mu,\psi} = \begin{cases} \exp\Big\{ -\Big(1 + \xi\frac{x-\mu}{\psi}\Big)^{\frac{-1}{\xi}}\Big\},\ 1 + \xi\frac{x-\mu}{\psi} > 0\ , \xi \neq 0; \\ \exp\Big\{ -\exp\Big(-\frac{x-\mu}{\psi}\Big)\Big\},\ \xi = 0 \end{cases} \tag{2.7}$$

in which ξ is called shape parameter, μ location parameter and ψ scale parameter. The set $\theta = (\xi, \mu, \psi)$ can be called the set of model parameters.

For $\xi > 0$, the distribution is called heavy-tailed with polynomial decay and infinity right endpoint (Fréchet); for $\xi = 0$, it is called exponential (Gumbell); and for $\xi < 0$, it is called light-tailed with finite right endpoint (Weibull). Figure 2 illustrates how the process of fitting values of a distribution tail works for GEV distributions.



(a) Time series of monthly values with annual block size.



(b) GEV distribution with different ξ (shp) values. The parameters ψ (scl) and μ (ctr) are 1 and 0, respectively. Extracted from NCAR (2019).

Figure 2: (a)The blue dots in the time series are selected by the Block Maxima approach and represent the tail of the distribution. (b)Then, the blue dots are modeled in a GEV distribution that can be Fréchet, Gumbell or Weibull.

Maximum Likelihood Estimation

Intuitively, the maximum likelihood method selects parameters that makes the observed data more likely.

Equation 2.7 corresponds to the standard parametric case of statistical inference and, therefore, can be solved by maximum likelihood. Suppose that the generalized distribution function H_θ has density function h_θ:

$$h_\theta(x) = \frac{1}{\sigma}\Big[1+\xi\Big(\frac{x-\mu}{\sigma}\Big)\Big]^{(-\frac{1}{\xi}-1)} \exp\Big\{-\Big[1+\xi\Big(\frac{x-\mu}{\sigma}\Big)\Big]^{-\frac{1}{\xi}}\Big\},$$

for $1+\xi\Big(\frac{x-\mu}{\sigma}\Big) > 0$. The likelihood function, then, based on data $X = (X_1, \ldots, X_N)$, is given by

$$L(\theta; X) = \prod_{i=1}^{N} h_\theta(X_i).$$

Let $\ell(\theta; X) = \ln L(\theta; X)$ be the log-likelihood function, in which:

$$\ell(\theta; X) = -N \ln \sigma - \sum_{i=1}^{N} \left[1 + \xi\Big(\frac{X_i - \mu}{\sigma}\Big)\right]^{-\frac{1}{\xi}} - \Big(\frac{1}{\xi} + 1\Big) \sum_{i=1}^{N} \ln \left[1 + \xi\Big(\frac{X_i - \mu}{\sigma}\Big)\right] \tag{2.8}$$

The maximum likelihood estimator for θ is:

$$\hat{\theta}_N = \arg \max_{\theta \in \Theta} \ell(\theta; X) \tag{2.9}$$

The numerical calculation of the maximum likelihood estimator, $\hat{\theta}_N$, for H_θ no longer represents a challenge since the existence of a FORTRAN algorithm published by Hosking (1985) and further investigated by Macleod (1989).

Return Level

The return level is the maximum amplitude, on average, after every t observations. For a GEV distribution, it is described by

$$\hat{z}_t = \begin{cases} \hat{\mu} + \frac{\hat{\sigma}}{\hat{\xi}}[(-\ln(1-t))^{-\hat{\xi}} - 1], & \hat{\xi} \neq 0 \\ \hat{\mu} + \hat{\sigma}[\ln(1-t)], & \hat{\xi} = 0 \end{cases}, \tag{2.10}$$

in which $\hat{\sigma}$, $\hat{\xi}$ and $\hat{\mu}$ are the parameters estimated by the maximum likelihood method.

2.1.3 Peaks Over Threshold

The Pareto (1898) distribution is one of the heavy-tailed distributions. So its generalization can be used to model extreme events.

The Generalized Pareto Distribution (GPD) can be defined by:

$$G_{\xi,\sigma}(x) = \begin{cases} 1 - (1 + \frac{\xi x}{\sigma})^{-\frac{1}{\xi}}, & \xi \neq 0, \\ 1 - e^{\frac{-x}{\sigma}}, & \xi = 0; \end{cases} \tag{2.11}$$

in which

$$\begin{cases} x \geq 0, & \xi \geq 0, \\ 0 \leq x \leq -\frac{\sigma}{\xi}, & \xi < 0; \end{cases} \tag{2.12}$$

for $\sigma > 0$ and $\xi \in \mathbb{R}$.

Let X be a random variable and a threshold u. Then, the random variable $X - u$ is the excess values and its distribution function, denoted by F_u, can be calculated by:

$$F_u = P(X - u \leq x | X > u) = \frac{P(X - u \leq x \wedge X > u)}{P(X > u)} = \frac{F(x + u) - F(u)}{1 - F(u)} \tag{2.13}$$

It follows that F_u can be approximated by a GPD:

Theorem 2 (Balkema & de Haan (1974); Pickands (1975))**.** *For a class of distributions, an appropriate positive function $\sigma(u)$ can be found such that:*

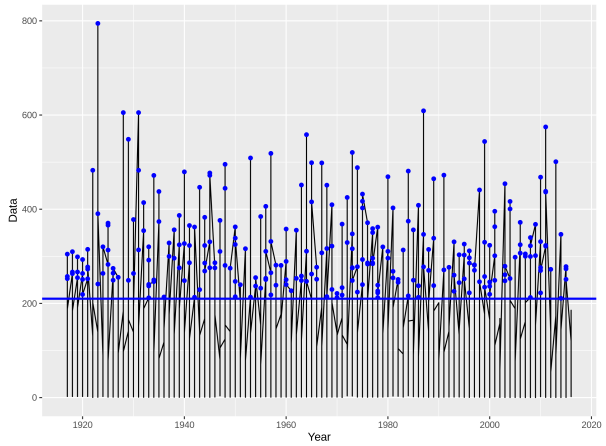
$$\limsup_{u \to x_F,\, 0 \le x < x_F - u} |F_u(x) - G_{\xi,\sigma}(x)| = 0$$

The class of distributions for which this theorem is valid includes most of standard distributions, i.e., Normal, Log-normal, Beta, Exponential, Uniform, etc.

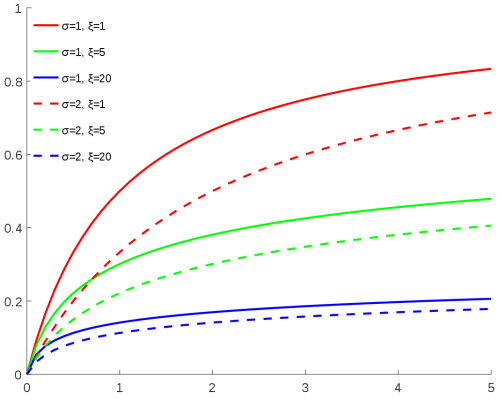
Based on this result, for a large value of u, the following approximation is possible:

$$F_u(x) \approx G_{\xi,\sigma}(x) \tag{2.14}$$

Thus, the GPD can be used to model distribution tails for data that exceed a threshold, as shown in Figure 3.



(a) Time series with exceedances above threshold $u = 220$ in blue.



(b) GPD distribution with different values of ξ, σ. Extracted from Wikipedia (2019).

Figure 3: (a) The blue dots in the time series are selected by the Peaks Over Threshold approach and represent the tail of the distribution. (b)The blue dots are modeled in a GPD distribution.

Maximum Likelihood Estimation

Assuming F is GPD with parameters ξ, σ, so that the density function is

$$f_{\xi,\sigma}(x) = \frac{1}{\sigma}\Big(1 + \xi\frac{x}{\sigma}\Big)^{-\frac{1}{\xi}-1}$$

Using the likelihood function, based on data $X = (X_1, \ldots, X_N)$,

$$L((\xi, \sigma); X) = \prod_{i=1}^{N} f_{\xi,\sigma}(X_i) \tag{2.15}$$

Then, the log-likelihood function is

$$\ell((\xi,\sigma);X) = -N\ln\sigma - \Big(\frac{1}{\xi}+1\Big)\sum_{i=1}^{N}\ln\Big(1+\frac{\xi}{\sigma}X_i\Big) \tag{2.16}$$

Thus, likelihood equations can be derived and solved numerically, obtaining the estimated parameters $\hat{\xi},\hat{\sigma}$.

Return Level

The return level is the maximum amplitude, on average, after every t observations. For a Pareto distribution, it is described as

$$\hat{z}_t = \begin{cases} u + \frac{\hat{\sigma}}{\hat{\xi}}[(t\hat{\lambda}_u)^{\hat{\xi}} - 1], & \hat{\xi} \neq 0 \\ u + \hat{\sigma}(t\hat{\lambda}_u), & \hat{\xi} = 0 \end{cases}, \tag{2.17}$$

in which u is the chosen threshold, $\hat{\sigma}$ and $\hat{\xi}$ are the scale and shape parameters, respectively, t is the number of observations and $\hat{\lambda}_u$ is the rate of observations above the threshold.

2.1.4 Graphical Approaches to Threshold Selection

Hill Estimator

The ξ parameter, also known as tail index, is determinant when inferring rare events, such as, the estimation of a high quantile non-usual (in finance, Value at Risk) or the dual problem of estimating the probability of exceeding a high value.

The Hill (1975) estimator is given by:

$$\hat{\alpha}_k = \frac{1}{\hat{\xi}_k} = \Big(\frac{1}{k}\sum_{j=1}^{k}\ln X_j - \ln X_k\Big)^{-1} \tag{2.18}$$

with $X_1,\ldots,X_n$ independent and identically distributed. It is important to notice that the estimator depends on the k-th upper order statistics, in which $k\to\infty$, $k/n\to 0$ with $n\to\infty$.

In this scenario, the upper order statistics are samples from the time series sorted in descending order.

The crucial aspect in using Hill estimator is the choice of k, that is directly connected to the threshold u. A value of u too high results in too few exceedances and, consequently, high variance estimators. For u too small, estimators become biased. To determine the value of k to be considered, it is advised to plot $(k,\hat{\xi}_k)$ and find a *plateau* region. This region is identified as the one with the values of $\hat{\xi}_k$ closer to the original value ξ.

In Figure 4, it is exhibited the Hill plot of Fort Collins (Colorado, USA) precipitation time series. Data were provided by the R package "extRemes" Gilleland (n.d.). In this plot, an adequate choice of threshold is in the region delimited by the 774 and the 873 order statistics. It is also worth mentioning that the two red lines are the confidence interval with $p = 0.95$.

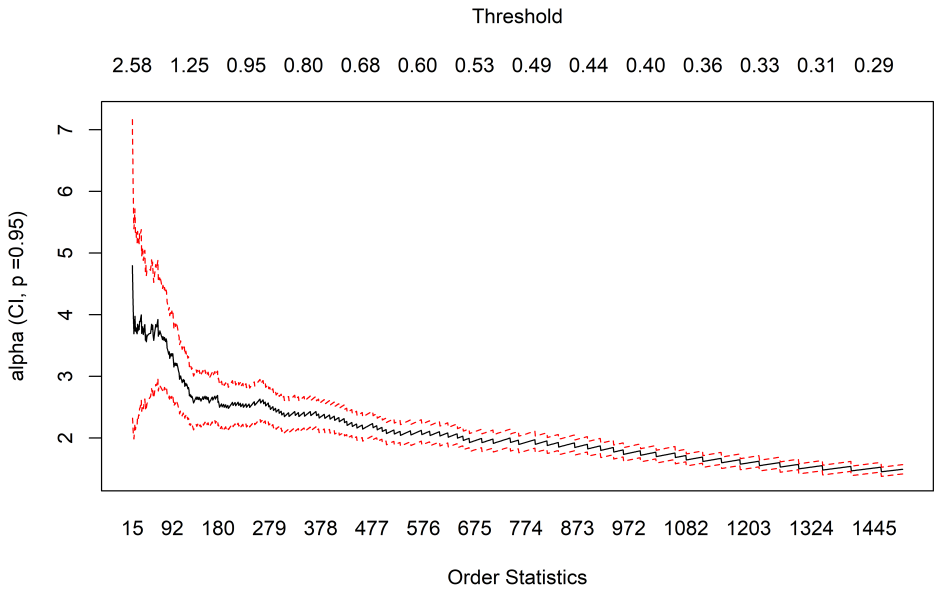


Figure 4: Hill plot of Fort Collins precipitation data.

Mean Excess Plot

Definition 1 (Mean excess function)**.** *Let X be a random variable with right endpoint x_F, then*

$$e(u) = E(X - u|X > u),\ 0 \leq u \leq x_F \tag{2.19}$$

is called mean excess function of X.

In the previous definition, the right endpoint concept was applied. Therefore,

$$x_F = sup\{x \in \mathbb{R} : F(x) < 1\}.$$

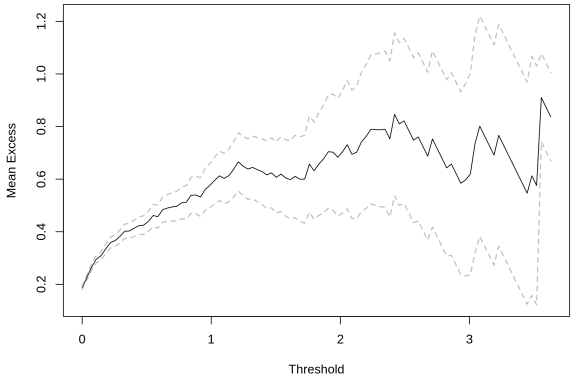
being F the distribution function.

The mean excess function exerts an important role due to the fact that, for variable u, it is linear in the GPD case.

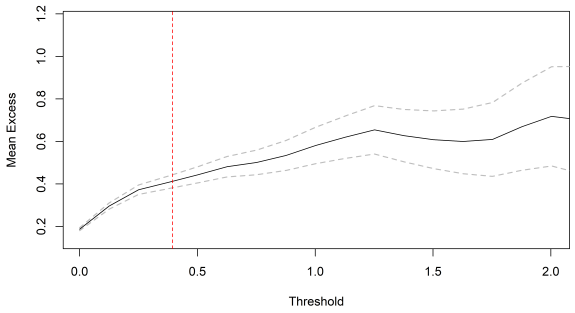
$$e(u) = \frac{\sigma + \xi u}{1 - \xi} \tag{2.20}$$

It is possible to verify in what region the mean excess function is linear for u, by calculating it for different values of threshold u and plotting the results. In this region, then, the approximation 2.14 is considered to be reasonable.

To find an optimal threshold u, in Figure 5b, a linear region must be searched and the starting point of the linear part represents the optimal threshold. In this case, the chosen threshold is $u = 0.395$, highlighted by the red horizontal line.



(a) Mean Excess plot for Fort Collins precipitation data.



(b) Mean Excess plot pruned with threshold $u = 0.395$ highlighted.

Figure 5: Example of mean excess plot.

2.1.5 Applications

Extreme Value Theory has been widely applied in countless fields to assess risk and predict the probabilities of extreme events. Here, a focus on applications of EVT in climate areas is provided.

EVT was employed to describe maximum monthly distributions of heavy precipitation in a certain location in Towler *et al.* (2010). The model presented also considered that there is non-stationarity, i.e., no concurrent information that indicates climate change needed to be used. With the statistics provided by the EVT application, it was possible to reconstruct flow quantiles and to project it to the year 2100. The paper also extended the analysis to changes in the quality of the water. Results show that, in the case study location, it will be an increase in the variability and magnitude of streamflow extremes and an increase in risk of turbidity exceedance was also quantified.

Another application of EVT was made by Cooley (2009), which produced a commentary of another paper about how slowly changing climate could affect the frequency of extreme events. The main objective was to discuss the advantages of an EVT approach and review techniques that were already used to describe the impact of climate changes in extreme phenomena. An analysis of temperatures of central England was also done, comparing a time-varying model with a stationary one.

In Naveau *et al.* (2005), three case studies were presented to show that EVT can provide a solid foundation when considering the uncertainty associated with extreme events. One of these studies focused on characterizing magnitudes of large volcanic eruptions, and it is shown that the effects of volcanic activity in climate should be modeled by a heavy-tailed distribution.

2.2 Model selection in machine learning

For each chosen u we can generate a model fitted using Maximum Likelihood Estimation. When the training phase ends, it is necessary to check the accuracy of the model when predictions are made. This process encompasses evaluating the quality of the model and selecting the one that performs the best in unseen data, thus exhibiting maximal generalization capability (Bishop, 2006; Hastie *et al.* , 2001).

When testing the effectiveness of the model, we look for a method that uses data in the best possible way when training and also assessing the performance. Two types of validation are the most common: holdout and k-fold cross-validation — the first consists of splitting the dataset into two sets: training and test. The training set is used to fit the model, and the test set is used to see how well the model performs on unseen data.

The second randomly splits the dataset into k different folds. One of the folds is used as the test set, and the rest is used as a training set. The model is trained on $k-1$ folds and tested on the test fold. The process is repeated until each fold was used once as the test set. Usually, the k-fold method results in a less biased model, since every sample appears in the training and test set at least once. The disadvantage here is the necessity of determining k and the existence of k learning models at the end. With $k = N$, in which N is the number of samples, the estimator is approximately unbiased for the expected prediction error, but can have high variance due to the N training sets being similar to each other. On the other hand, with $k = 5$ or $k = 10$, common values found in literature, the variance is lower, but bias can be a problem, depending on how the performance of the method differs with the size of the training set. Overall, the last proposal is recommended as a good compromise (Breiman & Spector, 1992; Kohavi, 1995). Figure 6 presents a graphical representation of the partition policy of the two types of validation.

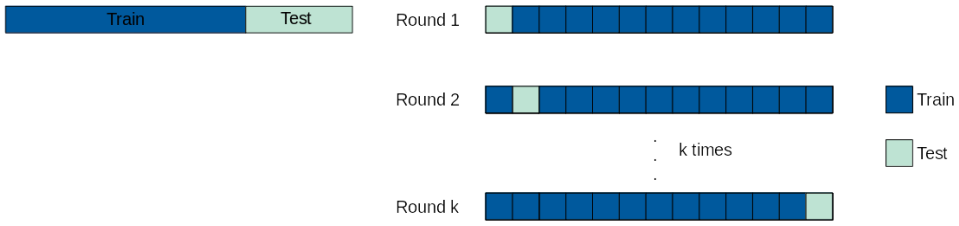


Figure 6: Comparison of the data preprocessing promoted by holdout and k-fold cross-validation.

In Nested Cross-Validation (Bergmeir & Benítez, 2012), an outer loop will train each time window with optimal parameters and then average each window's test error. An inner loop will tune hyperparameters by training a subset and validating it. Two main methods are applied in nested CV: predict second half and day forward-chaining (Bergmeir & Benítez, 2012). In the first type, the first half of the data, which is split temporally, is assigned to the training set, while the latter half is the test set. The validation set size depends on the problem. However, it is

always chronologically subsequent to the training set. The second type is based on rolling-origin-recalibration and each day is considered as the test set and all previous data is assigned to the training set. This method will be the main basis for the proposed method of Chapter 3.

2.2.1 Applications

In Kohavi (1995), a review of accuracy estimation methods and a comparison of the two most common ones — bootstrap and cross-validation — is given. A large-scale experiment — over half a million runs of C4.5 and a Naive-Bayes algorithm — is executed to estimate the effects of different parameters on these algorithms on real-world datasets. The results obtained indicate that for real-world datasets that are similar to the ones used, the best method to use for model selection is 10-fold stratified cross-validation.

Cross-validation for model selection can be applied in multiple areas. For example, in Sharma *et al.* (2017), the performance of several machine learning classifiers was assessed for the discrimination between the vegetation physiognomic classes, using the satellite-based time-series of the surface reflectance data. A set of machine learning experiments comprised of some supervised classifiers with different model parameters were conducted to assess how the discrimination of these classes varies with classifiers, input features, and ground truth data size. The performance of each experiment was evaluated using 10-fold cross-validation.

2.3 Multitask Learning

2.3.1 Single-Task Learning vs Multitask Learning

When facing a set of learning tasks, for example, predicting the forecast of distinct locations, the usual procedure is to learn each task individually, recombining the solution after this step. This reductionist approach ignores the fact that, given multiple learning tasks, a subset of those tasks may act as valuable sources of knowledge for each one of those tasks, so that the exploration of the relationship among multiple tasks may benefit performance. Consequently, that mechanism generates a faster and more precise learning process (Caruana, 1993).

In conclusion, this is the main difference between the two learning processes in machine learning: single-task and multitask. While the first is focused on learning each task in specific, using for that only data that is related to that task; the second integrates the knowledge of all tasks. Moreover, that integration act as a parallel of how human knowledge works. For example, if a child is taught to run, to jump, to walk, to estimate trajectories and to recognize objects, she/he probably easily learn how to play soccer. As much as these tasks are not the same in different contexts (run in soccer and run in a cinder track) some similarities allow the transference of knowledge or ability.

Thus, an MTL approach will use information contained in train signals of related tasks as inductive knowledge that will benefit multiple tasks. Two components are essential to multitask learning: the information shared and the relationship among tasks.

Stein's Paradox

A result that validates the principle that joint learning of related tasks can leverage performance when compared to learning each task individually is Stein's Paradox (Stein, 1956). This paradox states that, when three or more parameters are estimated simultaneously, there are estimators that, when combined, produce more accuracy, on average, than any other method that estimates these parameters separately.

Formally, let θ be a vector with $n \geq 3$ unknown parameters. To estimate these parameters, let X_i be the measure for each parameter θ_i, resulting in a vector X of size n. Suppose that $X \sim \mathbf{N}(\theta, 1)$ is a more intuitive way to estimate its corresponding parameters, that is,

$$\mathcal{L}_{\hat{\theta}}(\theta) = \mathbb{E}\{\| \theta - \hat{\theta} \|^2\} = \int (\hat{\theta}(x) - \theta)^2 p(x|\theta) dx. \tag{2.21}$$

In other words, the risk function measures the expected value of the estimation error. An estimator $\hat{\theta}$ is admissible if there is no other estimator $\tilde{\theta}$ with smaller risk. Stein proved that $\hat{\theta}$ is admissible for $n \leq 2$, but inadmissible for $n \geq 3$.

Spite of Stein's Paradox is considered as a premise for the hypothesis of multitask learning, since it works with unrelated random variables. The difference between the two approaches is also in the fact that MTL estimates the parameters of a certain task with unknown distribution, while in Stein's Paradox the variables follow a normal distribution.

2.3.2 Formulation

In machine learning, it is usual to minimize the empirical error:

$$\min_{\Theta} \mathcal{L}(\Theta) \tag{2.22}$$

such that Θ is the set of estimated parameters for the training samples, and $\mathcal{L}(\Theta)$ is the empirical cost in the training set, that measures the performance of the learning task in the training set.

Given a set S of n tasks, $S = S_1, \ldots, S_n$, the k-th training dataset is given by $(x_j^k, y_j^k)_{j=1}^{N_k}$, where $x_j \in \mathbb{R}^d$ is the input data and $y_j \in \mathbb{R}$ is the corresponding output, when a regression problem is solved, or $y_j^k \in \{0, 1\}$, when it is a binary classification problem. Consequently, the goal is to learn n parameter vectors $\theta_1, \ldots, \theta_n$ given that $f(x_j, \theta_k) \approx y^k$, $k = 1, \ldots, n$; $j = 1, \ldots, N_k$. Hence, the MTL cost function can be represented as follows:

$$\mathcal{L}(\Theta) = \sum_{k=1}^{n} \mathbb{E}_{(X_k, y^k) \sim p}[\ell(f(\theta_k, X_k), y^k)] = \sum_{k=1}^{n} \int_{\mathcal{X} \times \mathcal{Y}} \ell(f(x^k, \theta_k), y^k) dp(x^k, y^k) \tag{2.23}$$

In practice, the distribution p is unknown and only a finite number of i.i.d samples, N_k, is available. Hence, the total empirical cost can be described as

$$\hat{\mathcal{L}}(\Theta) = \sum_{k=1}^{n} \frac{1}{N_k} \sum_{j=1}^{N_k} \ell(f(x_j^k, \theta_k), y_j^k) \tag{2.24}$$

in which the loss function $\hat{\mathcal{L}}(\Theta)$ is generalized to aggregate the prediction cost of n tasks and Θ is a matrix containing n columns, being θ_k the k-the column of Θ.

2.3.3 Models of relationship among tasks

Capturing the structural relationship among tasks is a crucial step in multitask learning. As pointed in Caruana (1997), it is fundamental that information is shared only among related tasks. Notice that the task relationship impacts directly in the performance and, therefore, can cause *negative transfer* when unrelated tasks share information.

All tasks are related

In some models, it is assumed that all tasks are related and only the information that is shared among them is controlled. In this case, the matrix Θ is responsible for the sharing pattern.

One of the first approaches proposed was the one described in Caruana (1993), in which the relationship among tasks is defined in hidden units of a neural network and each task corresponds to an output in the output layer. In Evgeniou & Pontil (2004), the existence of a reference vector that is shared among tasks is explored, so that the regularization penalty is imposed to force the parameter vector of each task toward this reference vector. Next, Argyriou *et al.* (2006) suggest an approach that assumes that all tasks share a common set of features, so that a group sparsity penalty is imposed to the parameter vectors of the tasks. In Ji & Ye (2009), a trace norm minimization is incorporated to bias the optimization of the parameter vectors.

Nevertheless, these approaches do not consider the presence of outliers, since they consider that all tasks are related to each other or that tasks share a common structure. Aiming at a more flexible structural relationship among the tasks, Chen *et al.* (2011) proposed a robust MTL formulation, which decomposes the parameter vectors in two components: the first identifies the relationship among tasks using a low rank structure and the second identifies outlier tasks using a sparse grouping structure. On the other hand, in Gong *et al.* (2012) the parameter vectors are decomposed differently: one component capture common features in relevant tasks and the other identifies outlier tasks.

Tasks are related in cluster structures

Here it is presumed that not all tasks are related, although it is assumed that this relationship occurs in groups or clusters. Thus, the information is shared only among tasks that belong to the same group.

Initially, in Thrun & O'Sullivan (1996), a methodology was proposed to learn clusters of tasks using a pairwise relationship: distances are measures based on how well a task is fitted when using other task's model. In Xue *et al.* (2007), the approach consists in automatically identifying task structure without previously knowing the number of clusters, in which task similarity are learned based on a Dirichlet process. In Jacob *et al.* (2008), a convex formulation was presented, considering that the task group is not known a priori and the task's parameters in the same cluster are restrict to be similar.

The disadvantage of using an approach that treats structural relationship among tasks as clusters is that tasks in the same cluster are limited to have similar parameter vectors, which may not hold in all cases. Thus, a method that considers a graph structure characterized by weighted edges indicating how strongly tasks are related and how densely graphs are connected, may be more

attractive. In Zhou *et al.* (2011), a Laplacian regularization was brought to the MTL context, using a graph structural model. A limiting factor of these methods is that these models consider previous knowledge of the graph structure, information that is not always available.

To solve this limitation, some proposals were accomplished to learn the graph structure directly from data, along with the parameter vectors.

Task relationship is explicitly learned

Some proposals suggest that the dependency among tasks be incorporated in the learning process. Most of them is based on hierarchical Bayesian models, supposing some distribution in the task parameter matrix, Θ, that discovers information about task relationship.

In Zhang & Yeung (2010), task relationship is modelled by the covariance matrix of the tasks, so that its inverse is used in the task parameters learning phase. As a result, the inverse matrix must be calculated at each iteration.

In Widmer *et al.* (2010), tasks are leaf nodes in a hierarchical structure described by a tree and, thereby, two approaches were proposed to explore such hierarchy. In one of them, a top-down approach is applied to learn individually parameter vectors for all the nodes, in which each node is composed of the union of the data of tasks below in the hierarchy and a regularization penalty is imposed, imposing that the parameter vector of a node be similar to the parameter vector of its father node. In the other, all parameter vectors are simultaneously learned and the regularization penalty is imposed to the parameter vectors by a proximity measure derived from the hierarchy.

2.3.4 Applications

Multitask learning has given important contributions to multiple research areas. In this context, some climate applications of multitask learning are provided.

A co-evolutionary multitask learning algorithm is presented in Chandra (2017) to dynamic predict the wind intensity during the occurrence of a tropical cyclone as soon as the event takes place. In the algorithm proposed, each point is the cyclone data every six hours and works as a subtask. Therefore, when more points of data are given, more predictions can be made which makes the model dynamic and robust. When compared to conventional alternatives and single-task learning (evolutionary algorithm and cooperative neuro-evolution algorithm), significant performance gain occurred.

The problem of climate variables prediction considering global climate model outputs is treated by Gonçalves *et al.* (2015) as part of the performance evaluation of their MTL framework. Each geographical location is considered a task and their relations are encoded in a sparse graph. The graph structure is jointly learned with the task parameter vectors. It is important to notice that their framework is capable of discovering relations between tasks without using the geographical coordinates as input information. Results confirmed the better performance of the proposed approach against baseline methods and also shown that correlations among locations were correctly captured by the graph.

A multitask neural network (MTNN) was applied in a deep learning approach to predict heatwaves from longitudinal time series of climate factors in Iglesias *et al.* (2015). The MTNN

framework is a series of fully connected hidden layers, where the activations at a determined layer are a function of the previous layer. Experiments were conducted with a time series of 18 variables related to climate, such as temperature, atmospheric conditions, and solar radiation. Heatwaves were defined as the monthly maximum temperature exceeding the five-year return level for that month. For each location, four outputs were provided: (1) the next month's maximum temperature; (2) the last six month's future maximum temperature over the next five years; (3) the presence of a heatwave next month; (4) the presence of a heatwave in the last month over the next five years.

2.4 Hierarchical Clustering

Unsupervised learning has multiple techniques to model relationships among objects without depending on labeled data. As discussed in Section 2.3.3, unsupervised learning steps can be incorporated into the MTL context, by clustering similar tasks and sharing information only among cluster members.

When dealing with the task of assembling objects, clustering is a technique that groups similar data points so that points in the same group are more related to each other than points in other groups. Several methods have been proposed in the clustering literature, and they can be classified as hierarchical or partitional algorithms (Jain *et al.* , 1999).

Hierarchical approaches can be classified in two categories: agglomerative (*bottom-up*) and divisive (*top-down*). In the agglomerative approach, each object starts in its own cluster and an agglomerative strategy is recursively applied, creating larger clusters up in the hierarchy until all objects are in the same cluster or the given number of clusters is reached. On the other hand, in a divisive approach, all objects start in the same cluster and a divisive strategy is applied recursively, creating smaller clusters until each object is in its own cluster or a given number of levels is reached.

2.4.1 Agglomerative Hierarchical Clustering

Agglomerative Hierarchical Clustering can be formally described as: let $S = \{x_i\}_{i=1}^{N}$ be the set of N objects, where each of them is composed of s features. The goal is to establish the objects in a binary tree structure T, such that leaf nodes represent the objects of S, and internal nodes represent clusters of leaves directly below in the hierarchy. Initially, objects are assigned to its cluster T_i, and pairs of clusters (T_j, T_k) are recursively merged using a similarity method. Each merged cluster creates a new internal node, T_m.

Hierarchical clustering methods differ by the distance metric and the linkage criterion Berkhin (2006), and the choice of an adequate metric will influence the shape of clusters. Before starting the clustering process, it is necessary to determine the proximity matrix that contains the distance between each point using a distance metric. There are three main measures: single linkage, average linkage, and complete linkage.

- Single linkage: uses the smallest distance between clusters;
- Average Linkage: the algorithm uses the average distance between the clusters;

- Complete linkage: the algorithm merges each pair of clusters to minimize the maximum distance between them.

For the distance criterion, the most commonly used are Euclidean, Manhattan, and Maximum.

Given that, we explored traditional (HC) and non-traditional (NHC) hierarchical clustering. Given a distance metric $d(T_j, T_k)$ that states distance between clusters T_j and T_k, and given the clusters $\{T_1^l, \ldots, T_{n^l}^l\}$ in level l of the tree: **traditional clustering** chooses the pair of clusters T_j^l and T_k^l with lower distance $d(T_j^l, T_k^l)$ to be part of the next level $l+1$, the clusters T_j^l and T_k^l are removed and all other are preserved; the **non-traditional clustering** chooses the pair of clusters T_j^l and T_k^l with lower distance $d(T_j^l, T_k^l)$ to be part of the next level $l+1$, after that, it iteratively find another pair of clusters (with the lowest distance) among the clusters not yet selected; this method removes all previous clusters, except when there is an even number of clusters. The cluster left out remains to the next level.

For example, in Figure 7 the clustering process of $N = 4$ objects is illustrated for both ways of composing the hierarchical structure. In Figure 7a, the non-traditional hierarchical clustering is represented. The algorithm checks the lowest distance value in the pairwise performance matrix and adds objects 3 and 4 to the same cluster. Since objects 1 and 2 are the remaining pair, they are also merged into a cluster, concluding step 1. Then, in step 2, there is only one possible combination, that merges all objects into a cluster, and, therefore, this is the chosen combination and the algorithm is finished.

On the other hand, in Figure 7b, the traditional hierarchical clustering is shown. In step 1, the algorithm also checks the lowest distance value in the pairwise performance matrix. The pair that corresponds to this value is merged into a cluster. Next, in step 2, all combinations are calculated, including the cluster (3,4). The smallest distance indicates that object 1 should be merged into the cluster (3,4) and that is what happens. Finally, in step 3, only one combination is left, object 2 with the cluster (1,(3,4)), and the algorithm finishes its execution.

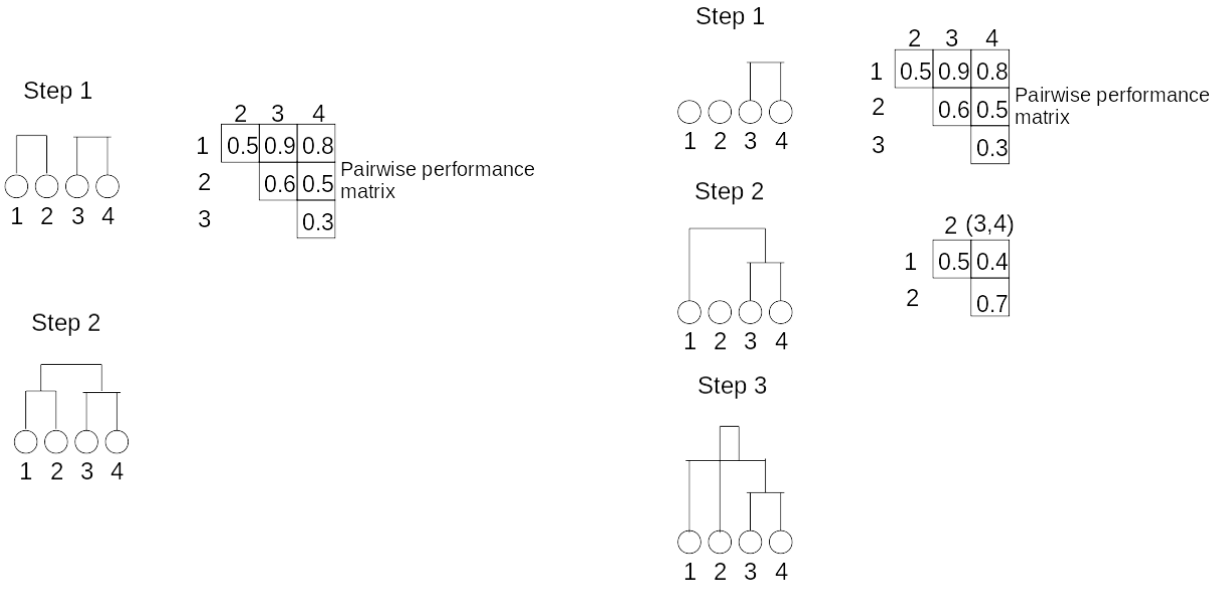
HC makes a more coherent exploration of the pairwise performance matrix, but NHC is computationally cheaper.

2.4.2 Applications

Clustering algorithms can be a very helpful tool when applied to the climate field since it allows the division of geographical areas into different climate districts. The analysis of the multiple clusters obtained by a clustering algorithm is useful to economic, agriculture and planning fields. For exploratory application, when the relations between different locations are unknown, hierarchical clustering is a recommended method.

An analysis of different climate zones of Turkey using temperature and precipitation data was performed in Unal *et al.* (2003). A comparison between hierarchical clustering methods with different distance criteria was also executed and seven different climate clusters were found.

In Stooksbury & Michaels (1991), another analysis was performed in southeastern US climate stations. In this case, hierarchical and non-hierarchical algorithms were combined in a two-step execution. The main objective was defining regions of climate homogeneity that should perform more robustly in climate impact models. These climate clusters may be more appropriate than



(a) Non-traditional hierarchical clustering (NHC).

(b) Traditional hierarchical clustering (HC).

Figure 7: Comparison between the two ways of composing the hierarchical structure.

the standard climate divisions, proposed by modifications of the agro-economic US Department of Agriculture crop report districts.

Chapter 3

Proposed Methodologies

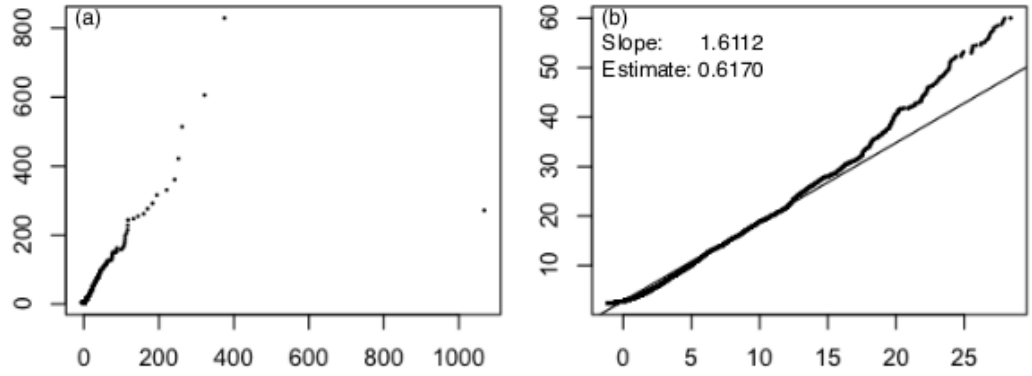
3.1 Motivation

Despite POTs being a better approach than GEV, mostly because it uses data more efficiently, this method is often neglected in fields such as climatology and meteorology (J. Scarrott & MacDonald, 2012). This occurs due to the necessity of analyzing multiple time series, aggravated by the absence of efficient and reliable methods to automatically select the threshold that acts as the best choice for each time series. Furthermore, the usual graphical methods are more popular. However, it lacks accuracy for requiring a very subjective choice.

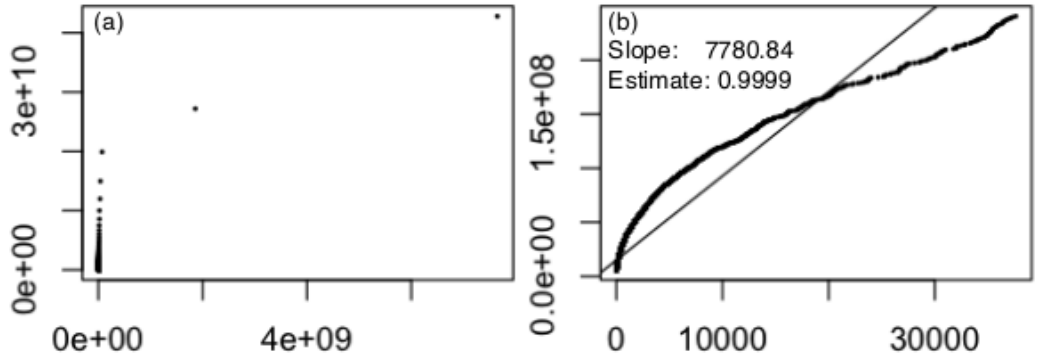
When using the mean excess plot, it is common the existence of more than one possible threshold, and discrepancies tend to arise (J. Scarrott & MacDonald, 2012). Another problem of the mean excess approach is that the analyst needs to trim away the plot for small and for too large values of the series. The situation here is similar to choosing the k upper order statistics of Hill Estimator, detailed in Section 2.1.4. Small values are governed by either the center or the left tail of the distribution. Too large values makes only a few samples be considered on the estimation of the parameters of the distribution. So two discretionary cuts need to be made. This situation is illustrated by Figure 8a. Finally, the analyst also needs to be convinced that $\xi < 1$, since for $\xi \geq 1$ the ME function does not exist, so the ME plot converges to a random curve, as shown in Figure 8b. More details about the disadvantages of the Mean Excess method is provided by Ghosh & Resnick (2009).

On the other hand, when using the Hill Plot it sometimes does not present a region of stability. When it occurs, the plot is called "Hill Horror plot", as shown in Figure 9. In this case, G_1 has a slowly varying tail, so it is possible to find a stable region when the Hill estimator is plotted. On the other hand, when plotting the Hill estimator of G_2, it is not possible to find a stable region and, therefore, and adequate threshold cannot be selected. The performance of Hill estimator can be very poor if the slowly varying function in the tail is far away from a constant (Embrechts *et al.* , 1997).

Aiming at a more robust estimation of the best threshold for each time series, an approach that is based on Hierarchical Clustering and Multitask Learning will be suggested here. This can be achieved by iteratively alternating between hierarchical multitask learning and the framework to train and test data, as indicated in Figure 10.



(a) Mean Excess plot of 50000 random variables. Entire plot (left) and order statistics 120-30000 (right).



(b) Mean Excess plot of 50000 random variables from Pareto distribution with $\xi = 2$. Entire plot (left) and order statistics (right).

Figure 8: Cases in which the ME plot can be very difficult to interpret and mislead the analyst to wrong choices of threshold. 8a shows the need to trim the plot; 8b shows that the plot has no linear region when $\xi \geq 1$. Extracted from Ghosh & Resnick (2009).

3.2 Model Assessment and Selection

To preserve the validity of the experiment, we first split the available dataset X_k, for the k-th task into training X_k^{tr}, and test X_k^{te} sets. The training set is used to fit and evaluate the candidate thresholds and the test set is used, after we found the most appropriate threshold u_k^*, to report the performance of the model. The proportion of this division is 80%/20%.

To evaluate a set of thresholds M_k for the k-th task in the training phase, we used an approach similar to Day Forwarding-Chaining, already presented in Section 2.2. This procedure consists of subdividing the training dataset into sliding windows, $X_k^{tr}[w]$. Each window has 20% of the training dataset. Then, the window dataset is used to fit a Pareto Distribution (GPD), using the MLE method and a threshold $u_k^i \in M_k$. The estimated parameters, (ξ_k^w, σ_k^w), are used to calculate

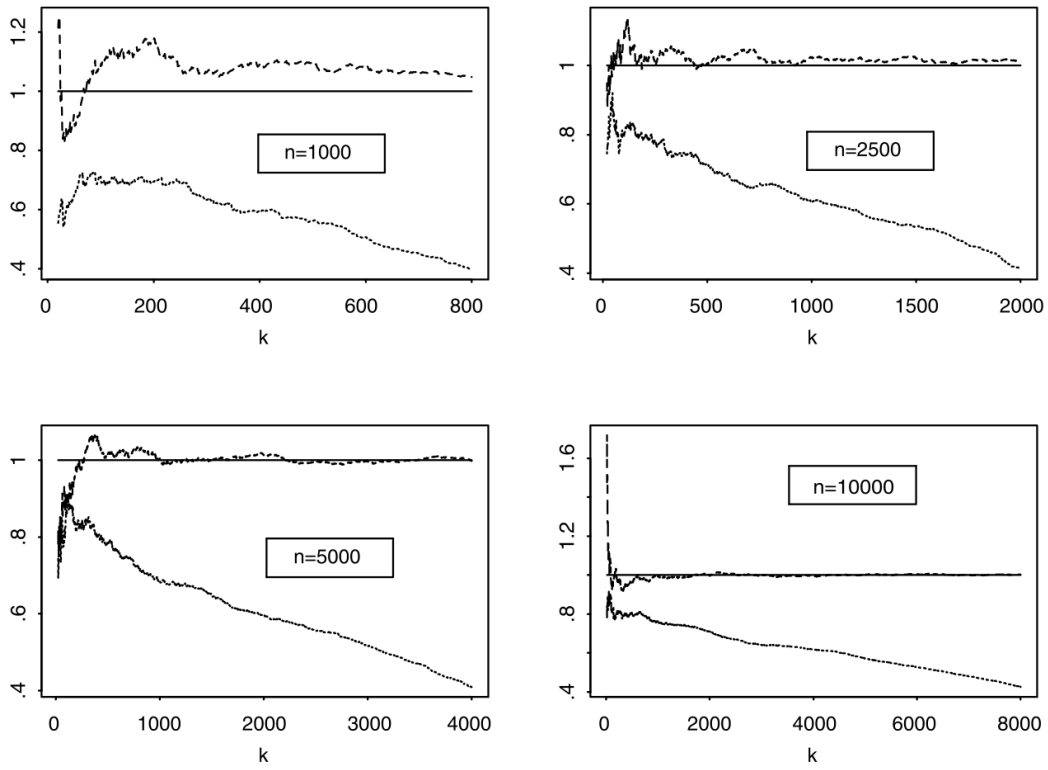


Figure 9: "Hill horror plot". The Hill estimator of n iid samples with distribution tail $G_1 = 1/x$ (curve above) and $G_2 = 1/x \ln x$ (bottom line). The solid line corresponds to when the estimator is 1. Extracted from Embrechts *et al.* (1997).

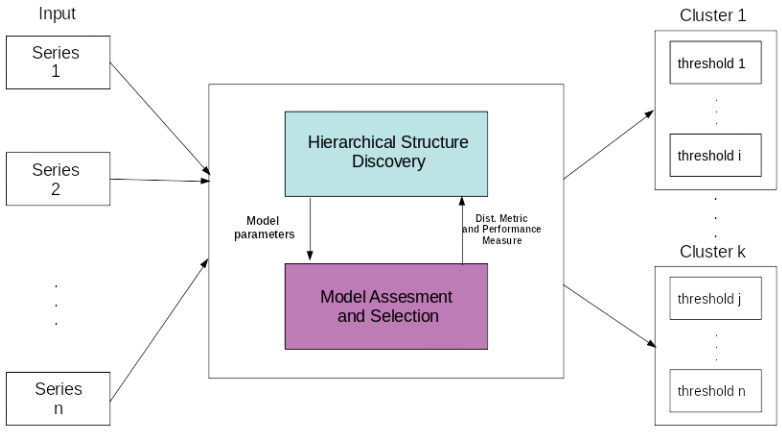


Figure 10: Flowchart of the proposed method.

the annual return levels, two years later ($T = 24$), as the following equation indicates.

$$\hat{x}_t = \begin{cases} u_k^* + \frac{\hat{\sigma_k^w}}{\hat{\xi_k^w}}[((T+t)\hat{\lambda}_u)^{\hat{\xi_k^w}} - 1], & \hat{\xi_k^w} \neq 0 \\ u_k^* + \hat{\sigma_k^w} \ln[((T+t)\hat{\lambda}_u)], & \hat{\xi_k^w} = 0, \end{cases} \tag{3.1}$$

in which $\hat{\lambda}_u$ is the rate of observations above the threshold and t is the number of observations that range from 1 to 12.

Later, the annual mean error of the fit was calculated by the monthly difference between the prediction $\hat{x}_t$ and the real value two years ahead, guiding to:

$$e = \frac{1}{12}\sum_{t=1}^{12}|x_{T+t} - \hat{x}_t|. \tag{3.2}$$

Afterwards, the validation window was translated in one year, and the whole process is repeated. After all training data is fitted, the l-infinity norm of the error was calculated as a performance measure. Non-extremal samples occur with higher frequency and, consequently, have more impact in usual measures, such as the mean squared error (MSE). Alternatively, the l-infinity norm measures the most significant difference, which probably happens when an extreme sample is estimated, and minimizing the worst case, the best threshold to estimate extremes are returned.

$$\|l\|_\infty = \max_i |e_i| \tag{3.3}$$

This process is repeated for every single threshold u_k in M_k. Then, the threshold u_k^* that generated that minimum $\|l\|_\infty$ is used in the test set. All data from the training dataset is fitted in a GPD by the ML estimator, and the parameters returned are used to calculate the return levels for each year of the test set. Subsequently, the return values are compared to the real values of the test set, and the l-infinity norm of the annual mean error is obtained. This value will be used as a performance measure of the internal nodes of the tree, which is constructed by the Hierarchical Multitask Learning Framework. At the end of the execution, the best performance nodes are returned for each task.

The whole process of training, validation and testing is represented in Figure 11.

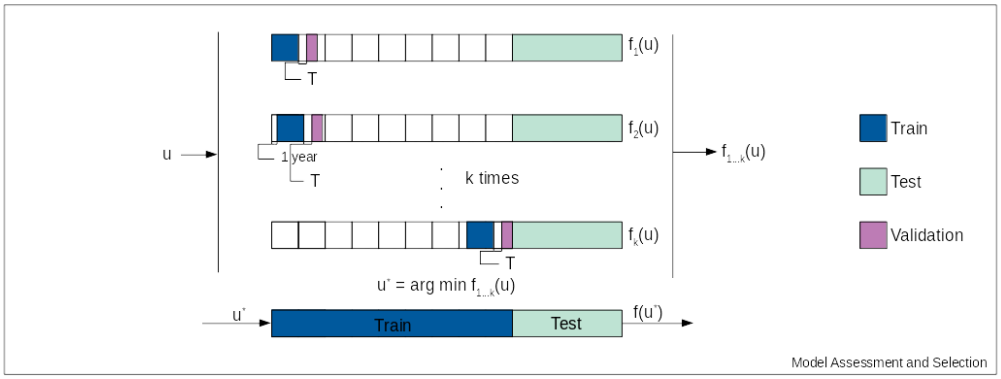


Figure 11: [Best view in color] Nested Cross-Validation proposed.

In Figure 12, the set subdivision is the one detailed in Figure 11. The inner and outer loop are the ones explained in Section 2.2. While the inner loop consists in training and validating all thresholds, the outer loop selects the best threshold for a determined time series and checks its performance when comparing with the test set.

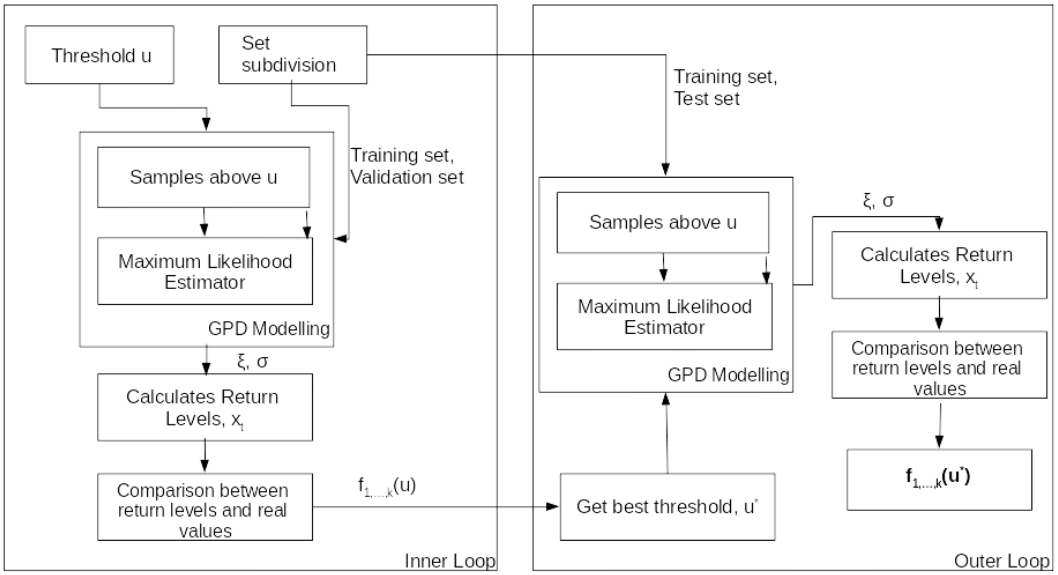


Figure 12: Fluxogram of the inner and outer loop based on nested cross-validation.

3.3 Hierarchical Multitask Learning Framework

In this work we explored the traditional and non-traditional clustering methods (presented in Section 2.4) to jointly search for the parameter u among the learning tasks. Each task i (a time series with extreme events) represents a leaf node $T_i^0 \equiv \{i\}$ in level 0. At any level l of the clustering tree, a new possible cluster T_m^l consists of the set of tasks belonging to the clusters T_j^l or T_k^l. Given that, it is possible to define a set (better explained below) of thresholds $M_{T_m^l}$ to that new cluster. The distance $d(T_j^l, T_k^l)$ consists of the mean performance $||l||_\infty^{T_m^l} = \sum_{i \in T_m^l} \frac{1}{|T_k^l|} ||l||_\infty^{u^*_{i \in T_m^l}}$ for all tasks. The performance for each task i is found by the best threshold $u^*_{i \in T_m^l} \in M^{T_m^l}$ in the set of thresholds $M_{T_m^l}$ determined by the cluster T_m^l.

The information sharing between tasks happens when the range of possible thresholds $M_{T_k^l}$ is defined. Besides, when two or more tasks are in the cluster, the range is defined by merging all time series into one. However, each task is trained and tested individually. Three possible alternatives are presented to select this range and they were also used in the single-task configuration:

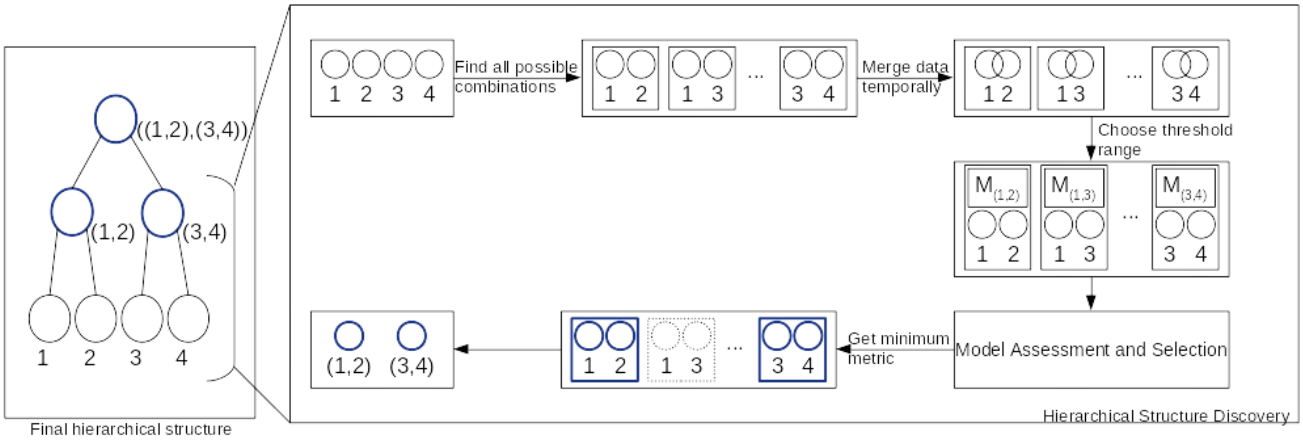
1. R1: The range is composed of data samples corresponding to the last quartile, i.e., all series values that are above the 75th percentile when varied in one unit;

2. R2: The range is composed of 50 data samples above the 75th percentile, similar to the first possibility, but the step is 0.5 instead of 1;

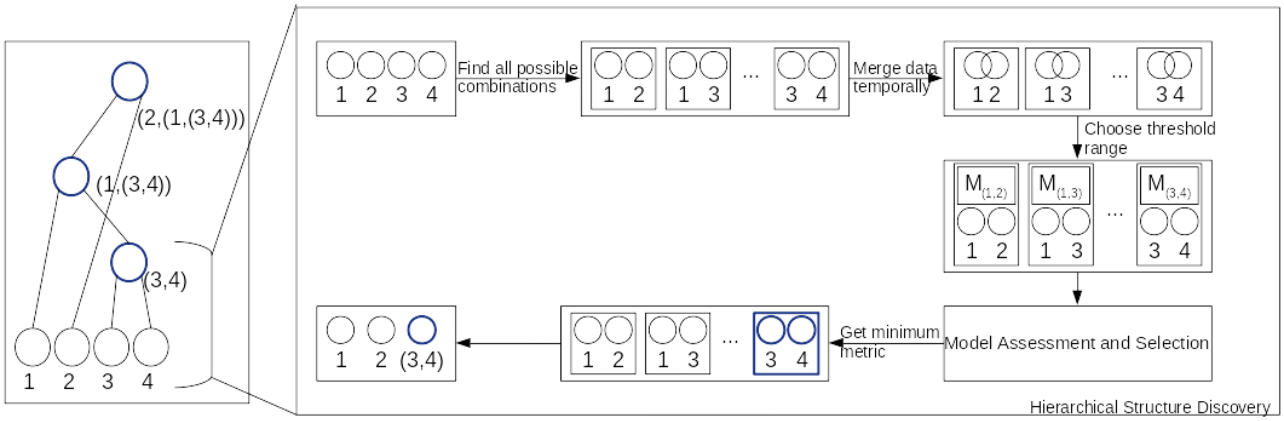
3. R3: The range is composed of all data samples greater than the corresponding value of the 75th percentile.

The 75th percentile was chosen because it represents the beginning of the tail of the distribution, as pointed in Bader *et al.* (2018).

As a result of more combinations to be evaluated, the traditional approach has a computational time considerably higher than the non-traditional one.



(a) Non-traditional hierarchical approach.



(b) Traditional hierarchical approach.

Figure 13: A complete journey through the first agglomerative step when performing the hierarchical structure discovery (the number of tasks is N = 4).

Figure 13 illustrates how the operations in the Hierarchical Structure Discovery module are done. Both approaches, non-traditional and traditional, were considered.

Chapter 4

Results

In this chapter, the experimental setup is presented as well as a discussion on the obtained results. First, a brief description of the dataset is given. Then, the obtained results when evaluating the performance of the proposed method comparing to the graphical one were described. Finally, a test with all time series from the dataset is executed to evaluate the benefits of multitask learning.

4.1 Dataset description

For the experiments, a monthly precipitation dataset provided by GPCC (Global Precipitation Climatology Centre) (Schneider *et al.* , 2016) was used. This dataset has the spatial coverage of 2.5° latitude × 2.5° longitude, in which each geographical location corresponds to a time series precipitation, and this grid was sliced to cover only South America. The locations of each time series is presented in Figure 14. Besides including Brazil, another motivation to define South America as the case study is the challenging scenario brought by its very diverse climate. Moreover, the temporal coverage used was from 1917 to 2016.

In addition to the slicing process, only the series that did not contain missing values were considered, since data imputation can lead to biased parameter estimation and the total number of series was considered adequate to perform the experiments here explained.

4.2 Comparison between graphical and the proposed automatic approaches

Aiming at comparing the results of the graphical (GM) and the proposed automatic method, 20 precipitation time series were considered. All series contain at least one extreme and they were selected so that at least five geographical regions from Brazil were covered. Then, the mean excess graphic of each time series was plotted in order to select an adequate threshold. As described in Section 2.1.4, a linear region was searched and the corresponding threshold was the one selected. After this procedure, the threshold was used to train 80% of the data and the return levels obtained were compared to the real values of the test set, which corresponds to the final 20% data of the

Figure 14: Geographical locations of data.

time series. The l-infinity norm was calculated.

Given the two possible ways to construct the hierarchical structure, two types of tests were executed: one with the non-traditional approach (NHCMTL) and the other with the traditional approach (HCMTL). In both of them, the automatic multitask algorithm was executed with all 20 series, following the method described in Chapter 3. Additionally, the single-task method (STL) consisted in running the proposed cross-validation, already explained in Section 3.2, with all thresholds from the chosen range. Also, the ranges R1, R2 and R3 from Section 3.3 were applied here. The results are presented in Table 1.

To detect differences between the experiments with each algorithm, a Friedman Test (Friedman, 1937, 1939, 1940) was applied in the results of Table 1 with threshold 0.01. A Friedman Test consists in a non-parametric statistical test that is used to detect differences in treatments among multiple models across multiple experiments. A Post-hoc Finner test is used to point which experiment is better. For the NHCMTL, the Friedman test found statistical significance, but the Finner post-hoc with the same threshold did not.

The same tests were applied to the HCMTL algorithm. In this case, statistical significance were found in both tests, and the Finner post-hoc indicated that HCMTL-R3 is better than all the contenders, including the graphical method. The same test pointed that the graphical method and STL are equivalent but worse than all multitask approaches proposed here.

Table 1: Comparison between the l-infinity norms obtained by the graphical method (GM), single-task learning algorithm (STL) and the proposed algorithms (NHCMTL and HCMTL).

ID	GM	STL - R1	NHCMTL - R1	HCMTL - R1	STL - R2	NHCMTL - R2	HCMTL - R2	STL - R3	NHCMTL - R3	HCMTL - R3
s1	257,88	391,73	**224,03**	252,97	391,73	252,97	252,97	401,94	252,95	252,95
s2	388,12	291,50	249,47	256,69	288,70	248,36	262,30	243,15	243,15	**187,80**
s3	57,10	26,93	26,93	26,93	26,60	26,60	26,60	**26,59**	**26,59**	**26,59**
s4	171,46	235,63	227,37	**162,44**	235,63	213,46	**162,44**	236,14	229,57	162,71
s5	303,04	195,29	195,29	**176,93**	197,61	197,61	197,61	198,00	198,00	198,00
s6	39,97	25,41	25,41	25,41	25,41	**25,14**	25,41	33,31	33,31	33,31
s7	**247,95**	443,942	250,65	252,97	485,77	252,97	252,97	484,53	252,95	252,95
s8	211,04	228,70	228,70	216,82	228,70	195,94	191,29	218,93	**123,91**	207,53
s9	56,53	65,37	65,37	65,37	65,37	65,37	65,37	65,35	65,35	**53,24**
s10	224,43	276,46	249,47	216,82	276,46	237,61	237,61	276,19	273,85	**207,53**
s11	243,57	207,30	**93,40**	128,36	216,88	93,95	149,76	217,34	149,27	116,79
s12	336,68	455,49	269,66	220,95	455,49	271,54	220,86	455,40	251,92	**220,70**
s13	126,18	**37,44**	**37,44**	**37,44**	40,17	40,17	40,17	40,92	40,92	40,92
s14	277,69	322,97	269,66	244,78	328,55	271,54	**191,29**	332,14	247,09	247,09
s15	88,44	69,08	69,08	69,08	69,08	69,08	69,08	69,15	68,99	**64,69**
s16	258,42	362,32	250,65	244,78	353,51	**237,61**	**237,61**	353,13	247,09	247,09
s17	85,93	48,92	48,92	48,92	48,92	48,92	48,92	**48,87**	**48,87**	**48,87**
s18	196,59	324,50	224,03	**162,44**	324,50	195,94	**162,44**	323,92	251,92	162,71
s19	51,80	34,75	34,75	34,75	**34,30**	**34,30**	**34,30**	40,74	40,74	40,74
s20	331,71	246,74	246,61	246,74	310,56	248,36	243,45	309,84	**123,91**	243,39

In Table 2, the rank of each algorithm is listed, so the performances can be compared.

Table 2: Obtained ranking for the methods under comparison.

Graphical	STL-R1	NHCMTL-R1	STL-R2	NHCMTL-R2	STL-R3	NHCMTL-R3
4.35	4.7	3.325	4.95	3.125	4.8	2.75
Graphical	STL-R1	HCMTL-R1	STL-R2	HCMTL-R2	STL-R3	HCMTL-R3
4.85	4.85	2.95	5.05	2.8	5.0	2.5

In the first part of Table 2, the methods compared were the graphical, STL and NHCMTL; while in the second part the approaches compared were the graphical, STL and HCMTL.

First of all, it is essential to notice that graphical and STL are statistically equivalent, thus indicating that the automatic approach, even in a single task configuration, is capable of achieving a similar result without the need of an expert. More importantly, it is possible to conclude that multitask learning approaches are capable of improving the performance in extreme events, and even when NHCMTL approaches did not present statistical significance, it always produces better ranks. Although HCMTL presented better results than NHCMTL, HCMTL takes more processing time, mostly because the better combinations of tasks are chosen at each agglomerative step.

However, there are some cases in which the graphical method was better than the proposed method. That occurred mainly due to two reasons: the threshold selected by the graphical approach was not present in the possible range of the proposed method or the combination among tasks that would benefit the performance was not selected by the model since another combination

presented a smaller metric value.

In the cases in which the single task method outperformed the multitask approach, the reason was mostly because of the negative transfer phenomenon, i.e., two or more unrelated tasks were placed in the same cluster. This may happen since a task was not related to any other task present in the data set, mostly considering that climate areas in South America are very diverse. Indeed, the algorithm does not detect outlier tasks before the whole hierarchical structure is constructed.

A plot comparison of the real values, the return levels obtained by the graphic method and the return levels obtained by one of the algorithms proposed of the Series ID 4 are presented in Figure 15. The algorithm chosen is HCMTL-R3 due to its great accuracy when compared to the other proposed algorithms.

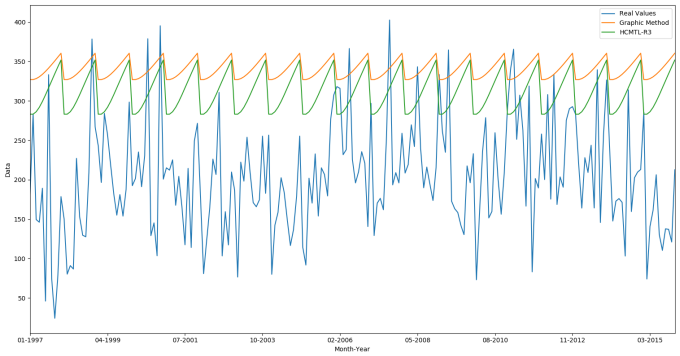


Figure 15: Comparison between the real values and the return levels generated by the graphic method and the ones generated by the HCMTL-R3.

The return levels generated by the HCMTL-R3 algorithm are below the graphic method due to the absolute value of the error being considered in the performance measure. Thus, both high values, corresponding to extreme precipitation, and low values, corresponding to droughts, influence the prediction. Also, it is important to notice that extreme prediction does not tend to follow the time series values, that is why the prediction curves do not have the same shape of the real values.

In Table 3, a comparison between the time execution of each algorithm is presented. This test was executed in a Intel(R) Core i9-9900K machine with CPU @ 3.6GHz and 16GB RAM. When analyzing the different possible ranges, the execution time is much higher for R3 since it uses all data above the 75th percentile. Thus, the number of possible thresholds and, consequently, the number of training phases increases with the number of combinations at each level of the hierarchical structure. In each combination, a task is incorporated into a cluster. Thereby, the range of possible thresholds includes all data above the 75th percentile of the concatenated data of all tasks clustered.

Another analysis is that HCMTL, in spite of being more accurate, takes more time to execute all the steps. The reason for the increase of execution time is that as the algorithm chooses only one combination to be a cluster per level of the hierarchical structure, there is an increase in both the number of levels and the combinations to be trained at each level.

Table 3: Comparison between the execution time of each algorithm.

Method	Time Spent (h:m:s)
NHCMTL - R1	00:28:07
NHCMTL - R2	00:53:20
NHCMTL - R3	04:37:08
HCMTL - R1	01:06:40
HCMTL - R2	02:16:36
HCMTL - R3	98:42:50

Fortunately, in our application involving precipitation extremes, even the higher execution times are reasonable and will not prevent the use of the proposed technique.

4.3 STL vs MTL

To measure the superiority of MTL methods over STL, another experiment was executed. In this one, 86 precipitation time series of the mentioned data set were used. The algorithm selected to train and test those series is the NHCMTL-R2, mostly because it has competent performance and takes less computational time than the other proposals with similar performance.

The Friedman Test was applied with 0.01 as a threshold, and the test found statistical relevance. The Finner post-hoc test pointed out the superiority of the multitask method. The l-infinity norms obtained are displayed on Appendix A.

Another essential feature of the multitask algorithm is the automatic proposal of a structural relationship among the prediction tasks. A map with the geographical locations and its estimated connections by the NHCMTL - R2 of all tasks is in Figure 16. Notice that the geographical location is not part of the provided information to the learning process, which implies that the coherent existence of dense connections between co-located points is something raised directly from the data-intensive methodology.

In Figure 16, the algorithm automatically discovered the coherent behavior between the precipitation series in north of Brazil, which contains the Amazon Forest, a region characterized by intense rainfall most of the year. Another region that was interconnected was the Brazilian Midwest, which has intense precipitation in the interval between October and March, and it is dry for the rest of the year.

However, some locations that were connected by the multitask algorithm does not have a similar rainfall regime, such as the tropical rainforest and an arid region, known in Brazil as northeastern “sertão". Nevertheless, these connections may have arisen because the difference between extremes and normal values is similar.

(a) Level 0: 43 edges.

(b) Level 1: 22 edges.

(c) Level 2: 11 edges.

(d) Level 3: 6 edges.

(e) Level 4: 3 edges.

(f) Level 5: 2 edges.

Figure 16: [Best viewed in color.] Relationship between tasks represented in a geographical map. Each figure shows a level of the hierarchical structure while in construction and the clusters in each level are in the same color.

Chapter 5

Conclusion

5.1 Concluding remarks

The overview of the Extreme Value Theory presented in Chapter 2 showed the importance of this field to predict extreme events whether they are in the realm of epidemiology, meteorology or even finance. The initial goal of this research was only in the final result of the fitted distribution, the return value, which allows the maximum amplitude prediction of the time series values in a given time observation. However, by studying the two main approaches to model the tail of the distribution and realizing that the one that is most indicated to model precipitation data is GPD, a new research opportunity was identified.

Usually, the threshold - value that separates time series data in exceedances - is graphically determined and, as demonstrated in Section 2.1.4 of a subjective nature and expert-dependent.

Therefore, an approach to solve the problems of subjectivity and expert dependency of the graphical models were proposed. The hierarchical clustering method is well-known in the multitask learning field as it is used to represent the task relationship that is directly learned from task data. The cross-validation is applied to select the model that obtains the best performance. Thus, all these strategies presented were combined and resulted in a framework to automatically select an adequate threshold for each task in a given group of tasks.

Then, to compare the presented method with the usual graphical method, two experiments were conducted. One of them consisted of 20 precipitation time series of South America. The second one was applied in 86 precipitation time series to determine if the multitask approach was more effective than the single task learning procedure. The proposed method was not only better than the graphical approach but was also superior to the single task counterpart, indicating that multitask learning can improve algorithm performance. The presented method will necessarily propose a structural relationship among tasks. Consequently, this structure can be used to analyze qualitatively the relationship among different time series, given their geographical location.

5.2 Future Directions

Despite of the promising results obtained until now, opportunities for further improvements were identified. Therefore, a list of future directions is provided:

- **Hierarchical Clustering:** Non-binary hierarchical clustering is a promising extension that will allow more than two tasks in the merging phase, producing rose trees. Therefore, the computational cost can decrease since the number of levels in the tree tends also to decrease.
- **Multivariate prediction:** Including other types of possibly correlated climate time series, such as wind speed, temperature and relative humidity, will also be pursued to further explore the knowledge transfer promoted by multitask learning. Thus, the relationship between tasks can be better identified, possibly leading to improved results.
- **Outlier Tasks:** Investigation of other clustering structures capable of removing outlier tasks before the training phase starts, thus avoiding unnecessary computational cost and negative transfer.

Bibliography

Aguiar, Nicole S., Raimundo, Marcos M., & Von Zuben, Fernando J. 2019. A multitask learning approach to automatic threshold selection in Pareto distributions. *[Submitted] Climate Dynamics*.

Argyriou, Andreas, Evgeniou, Theodoros, & Pontil, Massimiliano. 2006. Multi-Task Feature Learning. *In: NIPS*.

Bader, Brian, Yan, Jun, & Zhang, Xuebin. 2018. Automated threshold selection for extreme value analysis via ordered goodness-of-fit tests with adjustment for false discovery rate. *Ann. Appl. Stat.*, **12**(1), 310–329.

Balkema, A. A., & de Haan, L. 1974. Residual Life Time at Great Age. *The Annals of Probability*, **2**(5), 792–804.

Bergmeir, Christoph, & Benítez, José. 2012. On the use of cross-validation for time series predictor evaluation. *Information Sciences*, **191**(05), 192–213.

Berkhin, P. 2006. *A Survey of Clustering Data Mining Techniques*. Berlin, Heidelberg: Springer Berlin Heidelberg. Pages 25–71.

Bishop, Christopher M. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag.

Breiman, Leo, & Spector, Philip. 1992. Submodel Selection and Evaluation in Regression. The X-Random Case. *International Statistical Review / Revue Internationale de Statistique*, **60**(3), 291–319.

Caruana, Rich. 1993. Multitask Learning: A Knowledge-Based Source of Inductive Bias. *In: ICML*.

Caruana, Rich. 1997. Multitask Learning. *Machine Learning*, **28**(1), 41–75.

Chandra, Rohitash. 2017. Dynamic Cyclone Wind-Intensity Prediction Using Co-Evolutionary Multi-task Learning. *In:* Liu, Derong, Xie, Shengli, Li, Yuanqing, Zhao, Dongbin, & El-Alfy, El-Sayed M. (eds), *Neural Information Processing*. Springer International Publishing.

Chen, Jianhui, Zhou, Jiayu, & Ye, Jieping. 2011. Integrating Low-rank and Group-sparse Structures for Robust Multi-task Learning. *Pages 42–50 of: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '11. New York, NY, USA: ACM.

Coles, Stuart. 2001. *An introduction to statistical modeling of extreme values*. Springer Series in Statistics. London: Springer-Verlag.

Coles, Stuart G., & Tawn, Jonathan A. 1994. Statistical Methods for Multivariate Extremes: An Application to Structural Design. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **43**(1), 1–31.

Cooley, Daniel. 2009. Extreme value analysis and the study of climate change. *Climatic Change*, **97**(1), 77.

de Haan, L. 1990. Fighting the arch–enemy with mathematics. *Statistica Neerlandica*, **44**(2), 45–68.

Embrechts, Paul, Mikosch, Thomas, & Klüppelberg, Claudia. 1997. *Modelling Extremal Events: For Insurance and Finance*. Berlin, Heidelberg: Springer-Verlag.

Evgeniou, Theodoros, & Pontil, Massimiliano. 2004. Regularized Multi–task Learning. *Pages 109–117 of: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '04. New York, NY, USA: ACM.

Fisher, R. A., & Tippett, L. H. C. 1928. Limiting forms of the frequency distribution of the largest or smallest member of a sample. *Mathematical Proceedings of the Cambridge Philosophical Society*, **24**(2), 180–190.

Friedman, Milton. 1937. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *Journal of the American Statistical Association*, **32**(200), 675–701.

Friedman, Milton. 1940. A Comparison of Alternative Tests of Significance for the Problem of m Rankings. *The Annals of Mathematical Statistics*, **11**(1), 86–92.

Friedman, Milton M. 1939. A Correction: The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance.

Fukutome, S., Liniger, M. A., & Süveges, M. 2015. Automatic threshold and run parameter selection: a climatology for extreme hourly precipitation in Switzerland. *Theoretical and Applied Climatology*, **120**(3), 403–416.

Ghosh, Souvik, & Resnick, Sidney I. 2009. *A Discussion on Mean Excess Plots*.

Gilleland, Eric. *extRemes: Extreme Value Analysis*.

Gong, Pinghua, Ye, Jieping, & Zhang, Changshui. 2012. Robust Multi-task Feature Learning. *Pages 895–903 of: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '12. New York, NY, USA: ACM.

Gonçalves, A. R., Von Zuben, F. J., & Banerjee, A. 2015. A Multitask Learning View on the Earth System Model Ensemble. *Computing in Science Engineering*, **17**(6), 35–42.

Hallegatte, Stéphane, Hourcade, Jean-Charles, & Dumas, Patrice. 2007. Why economic dynamics matter in assessing climate change damages: Illustration on extreme events. *Ecological Economics*, **62**(2), 330 – 340. Special Section: Ecological-economic modelling for designing and evaluating biodiversity conservation policies.

Hastie, Trevor, Tibshirani, Robert, & Friedman, Jerome. 2001. *The Elements of Statistical Learning*. Springer Series in Statistics. New York, NY, USA: Springer New York Inc.

Hill, Bruce M. 1975. A Simple General Approach to Inference About the Tail of a Distribution. *Ann. Statist.*, **3**(5), 1163–1174.

Hosking, J. R. M. 1985. Algorithm AS 215: Maximum-Likelihood Estimation of the Parameters of the Generalized Extreme-Value Distribution. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, **34**(3), 301–310.

Hu, Huiling, & Ayyub, Bilal. 2019. Machine Learning for Projecting Extreme Precipitation Intensity for Short Durations in a Changing Climate. *Geosciences*, **9**(05), 209.

Iglesias, Gilberto, Kale, David C., & Liu, Yan. 2015. An Examination of Deep Learning for Extreme Climate Pattern Analysis.

J. Scarrott, C, & MacDonald, Anna. 2012. A review of extreme value threshold estimation and uncertainty quantification. *Revstat Statistical Journal*, **10**(03), 33–60.

Jacob, Laurent, philippe Vert, Jean, & Bach, Francis R. 2008. Clustered Multi-Task Learning: A Convex Formulation. *Pages 745–752 of:* Koller, D., Schuurmans, D., Bengio, Y., & Bottou, L. (eds), *Advances in Neural Information Processing Systems 21*. Curran Associates, Inc.

Jain, A. K., Murty, M. N., & Flynn, P. J. 1999. Data Clustering: A Review. *ACM Comput. Surv.*, **31**(3), 264–323.

Ji, Shuiwang, & Ye, Jieping. 2009. An Accelerated Gradient Method for Trace Norm Minimization. *Pages 457–464 of: Proceedings of the 26th Annual International Conference on Machine Learning*. ICML '09. New York, NY, USA: ACM.

Kohavi, Ron. 1995. A Study of Cross-validation and Bootstrap for Accuracy Estimation and Model Selection. *Pages 1137–1143 of: Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*. IJCAI'95. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Macleod, Allan J. 1989. [AS R76] A remark on Algorithm AS 215: "Maximum-likelihood estimation of the parameters of the generalized extreme-value distribution" (85V34 p301-310). *Applied Statistics*, **38**, 198–199.

McGovern, Amy, Elmore, Kimberly L., Gagne, David John, Haupt, Sue Ellen, Karstens, Christopher D., Lagerquist, Ryan, Smith, Travis, & Williams, John K. 2017. Using Artificial Intelligence to Improve Real-Time Decision-Making for High-Impact Weather. *Bulletin of the American Meteorological Society*, **98**(10), 2073–2090.

Naveau, Philippe, Nogaj, Marta, Ammann, Caspar, Yiou, Pascal, Cooley, Daniel, & Jomelli, Vincent. 2005. Statistical methods for the analysis of climate extremes. *Comptes Rendus Geoscience*, **337**(10), 1013 – 1022.

NCAR. 2019. *NCL: Basic Extreme Value Statistics*. [Online; accessed 15-November-2019].

Pareto, Vilfredo. 1898. *Cours d'Economie Politique*.

Pickands, James. 1975. Statistical Inference Using Extreme Order Statistics. *The Annals of Statistics*, **3**(1), 119–131.

Schneider, U., Becker, A., Finger, P., Meyer-Christoffer, A., Rudolf, B., & Ziese, M. 2016. *GPCC Full Data Reanalysis Version 7.0: Monthly Land-Surface Precipitation from Rain Gauges built on GTS based and Historic Data.*

Seneviratne, Sonia I., Nicholls, Neville, Easterling, David, Goodess, Clare M., Kanae, Shinjiro, Kossin, James, Luo, Yali, Marengo, Jose, McInnes, Kathleen, Rahimi, Mohammad, & et al. 2012. *Changes in Climate Extremes and their Impacts on the Natural Physical Environment.* Cambridge University Press. Page 109–230.

Sharma, Ram C., Hara, Keitarou, & Hirayama, Hidetake. 2017. A Machine Learning and Cross-Validation Approach for the Discrimination of Vegetation Physiognomic Types Using Satellite Based Multispectral and Multitemporal Data. *Scientifica*, **2017**.

Stein, Charles. 1956. Inadmissibility of the Usual Estimator for the Mean of a Multivariate Normal Distribution. *Pages 197–206 of: Proceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics.* Berkeley, Calif.: University of California Press.

Stooksbury, D. E., & Michaels, P. J. 1991. Cluster analysis of Southeastern U.S. climate stations. *Theoretical and Applied Climatology*, **44**(3), 143–150.

Thompson, P., Cai, Y. Reeve, D., & Stander, J. 2009. Automated threshold selection methods for extreme wave analysis. *Coastal Engineering*, **56**(10), 1013 – 1021.

Thrun, Sebastian, & O'Sullivan, Joseph. 1996. Discovering Structure in Multiple Learning Tasks: The TC Algorithm. *Pages 489–497 of: Proceedings of the Thirteenth International Conference on International Conference on Machine Learning.* ICML'96. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Towler, Erin, Rajagopalan, Balaji, Gilleland, Eric, Summers, R. Scott, Yates, David, & Katz, Richard W. 2010. Modeling hydrologic and water quality extremes in a changing climate: A statistical approach based on extreme value theory. *Water Resources Research*, **46**(11).

Unal, Yurdanur, Kindap, Tayfun, & Karaca, Mehmet. 2003. Redefining the climate zones of Turkey using cluster analysis. *International Journal of Climatology*, **23**(9), 1045–1055.

Valverde, M. C. 2017. The Interdependence of Climate and Socioeconomic Vulnerability in the ABC Paulista region. *Ambiente & Sociedade*, 09, 39 – 60.

Von Bortkiewicz, L. 1898. *Das Gesetz der kleinen Zahlen.* B.G. Teubner.

Widmer, Christian, Leiva, Jose, Altun, Yasemin, & Rätsch, Gunnar. 2010 (04). Leveraging Sequence Classification by Taxonomy-Based Multitask Learning. vol. 6044.

Wikipedia. 2019. *Generalized Pareto distribution — Wikipedia, The Free Encyclopedia.* [Online; accessed 06-November-2019].

Xue, Ya, Liao, Xuejun, Carin, Lawrence, & Krishnapuram, Balaji. 2007. Multi-Task Learning for Classification with Dirichlet Process Priors. *J. Mach. Learn. Res.*, **8**(Dec.), 35–63.

Zhang, Yu, & Yeung, Dit-Yan. 2010. A Convex Formulation for Learning Task Relationships in Multi-task Learning. *Pages 733–742 of: Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*. UAI'10. Arlington, Virginia, United States: AUAI Press.

Zhou, Jiayu, Yuan, Lei, Liu, Jun, & Ye, Jieping. 2011. A Multi-task Learning Formulation for Predicting Disease Progression. *Pages 814–822 of: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '11. New York, NY, USA: ACM.

Appendix A

Comparison between single task and multitask methods

Table 4: Comparison between the l-infinity norms obtained by the single task and the multitask learning algorithm.

ID	STL-R2	NHCMTL-R2
1	391,728	**239,657**
2	325,684	**203,731**
3	235,629	**222,401**
4	368,053	**276,543**
5	408,91	**276,543**
6	380,259	**262,208**
7	404,257	**226,757**
8	485,771	**230,071**
9	351,553	**206,357**
10	209,91	209,91
11	217,009	**212,207**
12	387,008	**267,829**
13	371,251	**276,543**
14	380,352	**245,844**
15	411,343	**276,543**
16	308,833	**225,312**
17	262,683	**246,471**
18	276,456	**229,906**
19	455,488	**276,543**
20	415,894	**245,844**
21	322,328	**267,829**
22	335,665	**267,829**

Continued on next page

Table 4 – continued from previous page

ID	STL-R2	NHCMTL-R2
23	300,907	**225,489**
24	289,64	**225,312**
25	289,536	**229,906**
26	298,792	**206,357**
27	330,909	**246,471**
28	333,529	**246,471**
29	353,515	**203,731**
30	355,862	**238,239**
31	302,92	**234,95**
32	378,287	**225,489**
33	361,361	**226,757**
34	392,752	**249,504**
35	355,708	**234,95**
36	283,085	**231,515**
37	268,706	**246,471**
38	334,281	**240,163**
39	468,999	**239,657**
40	417,599	**230,071**
41	496,495	**276,543**
42	390,398	**249,504**
43	404,096	**262,208**
44	370,51	**238,239**
45	295,138	**240,163**
46	344,769	**231,515**
47	324,349	**246,471**
48	325,292	**267,829**
49	239,54	**222,401**
50	288,697	**246,471**
51	197,607	197,607
52	200,5	200,5
53	310,561	**222,401**
54	179,263	**96,432**
55	118,957	**96,432**
56	228,698	**96,432**
57	65,5	65,5
58	74,151	74,151
59	62,947	62,947
60	46,926	46,926

Continued on next page

Table 4 – continued from previous page

ID	STL-R2	NHCMTL-R2
61	59,516	59,516
62	39,718	39,718
63	40,175	40,175
64	42,69	42,69
65	34,078	34,078
66	33,831	33,831
67	69,083	69,083
68	48,917	48,917
69	28,783	28,783
70	38,903	38,903
71	25,397	25,397
72	71,515	71,515
73	37,609	37,609
74	34,296	34,296
75	26,6	26,6
76	24,943	24,943
77	29,737	29,737
78	73,558	72,845
79	60,379	60,379
80	43,348	43,348
81	49,627	49,627
82	25,414	25,414
83	36,719	36,719
84	135,619	135,619
85	65,367	65,367
86	52,924	52,924