

# A Hybrid Pareto Model for Asymmetric Fat-Tail Data

Julie Carreau and Yoshua Bengio  
Dept. IRO, Université de Montréal  
P.O. Box 6128, Downtown Branch, Montreal, H3C 3J7, QC, Canada  
{carreau,bengioy}@iro.umontreal.ca

**Technical Report 1283**

Département d'Informatique et Recherche Opérationnelle

August 21st, 2006

## Abstract

We propose an estimator for the conditional density  $p(Y|X)$  that can adapt for asymmetric heavy tails which might depend on  $X$ . Such estimators have important applications in finance and insurance. We draw from Extreme Value Theory the tools to build a hybrid unimodal density having a parameter controlling the heaviness of the upper tail. This hybrid is a Gaussian whose upper tail has been replaced by a generalized Pareto tail. We use this hybrid in a multi-modal mixture in order to obtain a nonparametric density estimator that can easily adapt for heavy tailed data. To obtain a conditional density estimator, the parameters of the mixture estimator can be seen as functions of  $X$  and these functions learned. We show experimentally that this approach better models the conditional density in terms of likelihood than compared competing algorithms: conditional mixture models with other types of components and multivariate nonparametric models.

## 1 Introduction

The purpose of this paper is to introduce a new nonparametric model for conditional density estimation when the underlying distribution  $P(Y|X)$  is asymmetric and heavy-tailed. This task is meaningful in a number of application domains where one wishes to make predictions about a random variable  $Y$  (e.g., first and second moments or some relevant quantiles) given an observed  $X$ , when the distribution of  $Y$  given  $X$  can have fat tails, be multimodal or asymmetric.

Practical application domains where such distributions occur include financial and insurance modelling. In finance, estimating the predictive conditional distribution of the profit and loss (P&L) of a portfolio is central for portfolio and risk management. The usual risk measure is the so-called Value-at-Risk which is a quantile of the P&L distribution. Several authors have already provided strong evidence for the presence of fat tails in stock returns data [1]-[2]. In insurance applications, companies are interested in modelling the distribution of the claims given a client profile. For the auto insurance data used in the experiments described below, the profile includes information about the driver, the car, and the options selected by the insured for the insurance contract. In both application domains, the dimension of the input can be as much as hundreds.

In statistics, approaches for robust regression were put forward to deal with outliers [3]-[4]. Those could result from the fact that the data come from a fat-tailed distribution. Robust regression techniques involve decreasing symmetrically the weight of outliers; when the distribution is asymmetric, as it is the case in the applications we mentioned, those techniques are biased.

Regression involves estimating the conditional expectation  $E[Y|X]$ . We are interested in estimating the whole density  $p(Y|X)$  because besides  $E[Y|X]$ , other characteristics are relevant in many applications. In general, if one incurs a loss  $l(Y, X, d)$  when decision  $d$  is taken and  $Y$  and  $X$  are realized, then one should choose  $d$  to minimize the expected loss:

$$\int l(Y, X, d)p(Y|X)dY.$$

When  $l$  is not known precisely ahead of time, it is reasonable to look for an estimator  $\hat{p}(Y|X)$  of  $p(Y|X)$  that is close to the true one in the sense of the Kullback-Leibler (KL) divergence. However, since the true conditional density is unknown, one can consider the KL divergence with respect to the empirical distribution, which is equivalent to the conditional log-likelihood.

## 2 Extreme value theory

Extreme value theory [5] studies the behaviour of the maxima (or minima) of random variables. Each random variable  $X$  for which the maxima over several independent copies of that variable converges in distribution to a non-degenerate<sup>1</sup> random variable  $Y$  is said to belong to the maximum domain of attraction (**MDA**) of  $Y$ . The Fisher-Tippett theorem specifies the three possible limiting distributions for the maxima of random variables: the Fréchet, Weibull and Gumbel distributions. Hence, most distributions can be classified by stating to which domain of maximum attraction they belong.

The generalized Pareto distribution (GPD), whose distribution function is given in equation 1, appears as a good approximation to the tail of a distribution that belongs to any MDA (this is a fairly general condition). The tail parameter  $\xi$  determines to which MDA the tail belongs. When  $\xi > 0$ , the tail is heavy and belongs to the Fréchet MDA. When  $\xi = 0$ , the tail is light or moderately heavy, this is the Gumbel MDA. Finally, when  $\xi < 0$ , the tail is finite and comes from the Weibull MDA.

$$G_{\xi;\beta}(y) = \begin{cases} 1 - (1 + \frac{\xi}{\beta}y)^{-1/\xi} & \text{if } \xi \neq 0 \\ 1 - e^{-y/\beta} & \text{if } \xi = 0 \end{cases} \quad \begin{array}{l} \text{where } y \geq 0 \text{ when } \xi \geq 0 \text{ and} \\ 0 \leq y \leq -\beta/\xi \text{ when } \xi < 0 \end{array} \quad (1)$$

In order to use the GPD as an approximation for the tail of a distribution, we need to fix a threshold that defines where the tail begins. If the threshold is too high, very few points enter in the estimation of  $\xi$ , making the estimator subject to high variance. If the threshold is too low, the GPD approximation has a larger bias since the approximation of the tail is valid as the threshold goes to infinity. Some methods have been proposed for threshold selection but there is no consensus on which one to use.

## 3 Hybrid Pareto distribution

Since the GPD is only suited to model the tail of a distribution, we propose the **hybrid Pareto distribution** as a smooth extension of the GPD to the whole real axis. This new distribution is built by stitching

---

<sup>1</sup>A random variable is said to be degenerate if a single point has probability one.

a GPD tail to a Gaussian, while enforcing continuity of the resulting density and of its derivative. The threshold is thus defined implicitly as a function of the hybrid parameters. Let  $\alpha$  be the junction point (or threshold) and let  $f(y; \mu, \sigma)$  be the Gaussian density function with parameters  $\mu$  and  $\sigma$ ,  $g(y - \alpha; \xi, \beta)$  be the GP density with parameters  $\xi$  and  $\beta$  located above  $\alpha$ . The smoothness constraint on the density at  $\alpha$  means that  $f(\alpha; \mu, \sigma) = g(0; \xi, \beta)$  which gives:

$$\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\alpha - \mu)^2}{2\sigma^2}\right) = \frac{1}{\beta} \Leftrightarrow \exp\left(-\frac{(\alpha - \mu)^2}{2\sigma^2}\right) = \frac{\sqrt{2\pi}\sigma}{\beta} \quad (2)$$

Smoothness of the derivative of the density at  $\alpha$  means that  $f'(\alpha; \mu, \sigma) = g'(0; \xi, \beta)$ , which yields:

$$-\frac{(\alpha - \mu)}{\sqrt{2\pi}\sigma^3} \exp\left(-\frac{(\alpha - \mu)^2}{2\sigma^2}\right) = -\frac{(1 + \xi)}{\beta^2} \quad (3)$$

We plug equation 2 in equation 3 and we get that:

$$\frac{1 + \xi}{\beta} = \frac{\alpha - \mu}{\sigma^2} \quad (4)$$

We set  $\xi$ ,  $\mu$  and  $\sigma$  as the free parameters and we let  $\alpha$  and  $\beta$  be functions of these free parameters. We then solve equations 2 and 4 for  $\xi$ ,  $\mu$  and  $\sigma$ . For this, we make use of the Lambert  $W$  function: given an input  $z$ ,  $w = W(z)$  is such that  $z = we^w$ . We use a numerical algorithm of order four to find the zero of  $z - we^w$  [6]. The dependent parameters  $\alpha$  and  $\beta$  are obtained by the following formulae:

$$\beta(\xi, \sigma) = \frac{\sigma(1 + \xi)}{\sqrt{W\left(\frac{(1 + \xi)^2}{2\pi}\right)}} \quad \alpha(\xi, \mu, \sigma) = \mu + \sigma \sqrt{W\left(\frac{(1 + \xi)^2}{2\pi}\right)}$$

For  $\xi > -1$ , the hybrid Pareto density function is given by:

$$h(y; \xi, \mu, \sigma) = \begin{cases} \exp\left\{-\frac{(y - \mu)^2}{2\sigma^2}\right\} / (\gamma\sqrt{2\pi}) & \text{if } y \leq \alpha, \\ (1 + \xi(y - \alpha)/(\beta))^{-1/\xi - 1} / (\gamma\beta) & \text{if } y > \alpha \text{ and } \xi \neq 0, \\ \exp\{-y/\beta\} / (\gamma\beta) & \text{if } y > \alpha \text{ and } \xi = 0 \end{cases}$$

where  $\gamma$  is the appropriate re-weighting so that the density integrates to one and is given by:  $\gamma(\xi) = 1 + \frac{1}{2} \left(1 + \text{Erf}\left(\sqrt{W(z)}/2\right)\right)$ , where  $z = (1 + \xi)^2/(2\pi)$  and  $\text{Erf}(\cdot)$  is the error function  $\text{Erf}(z) = \frac{2}{\sqrt{\pi}} \int_0^z e^{-t^2} dt$ , which can be readily approximated numerically to high precision in standard ways.

## 4 Mixture models

We have integrated the choice of threshold into the estimation of the parameters of the hybrid Pareto and at the same time, we obtain a smooth density with an adaptable tail. In order to get a flexible estimator, we consider using hybrid Pareto densities as components of a continuous density mixture model. Using directly the GPD as a component of a mixture model would be impractical since the GPD is zero below the threshold that determines its location.

The mixture of Gaussians belongs to the MDA of the Gumbel distribution which encompasses distributions with light to moderately heavy tails. If the tail of the generative distribution is heavy, i.e. extreme observations can occur far away in the tail, good empirical results can often be obtained by considering a mixture of Gaussians in which one of the Gaussians has a very large  $\sigma$ , that serves to capture the points far away. One disadvantage of this approach is that the density model will only account for observed extremes and will still underestimate the density of the upper tail (this is specially true for small training sets). Another drawback is that by using symmetric components in the mixture, the lower tail tends to be overestimated.

## 5 Conditional mixture models

We build a conditional density estimator  $\hat{p}(y|x)$  based on the mixture model by modelling the mixture parameters for the density of  $y$  as functions of the input  $x$ . The estimator is given in equation 5 when using hybrid Pareto components.

$$\hat{p}(y|x) = \sum_{i=1}^m \pi_i(x) h(y; \xi_i(x), \mu_i(x), \sigma_i(x)) \quad (5)$$

Neural networks and linear or log-linear models are convenient classes of functions to compute the parameters of the output density, that is to implement the functions  $\pi_i(\cdot)$ ,  $\xi_i(\cdot)$ ,  $\mu_i(\cdot)$ , and  $\sigma_i(\cdot)$ , given an  $x$ , and they have been used successfully for similar tasks [7]. However any parametrized class of functions which can be trained using the gradient with respect to parameters can be used. This is because the estimation of this function is obtained through maximizing the mixture conditional log-likelihood. By increasing the number of hidden units, neural networks can in principle approximate any continuous function. The neural network output formulae are given in equation 6 and the resulting architecture is depicted in Figure 1.

The hidden unit activations,  $z_i$  are linearly combined to form the outputs of the neural network, the  $a_j$  in Figure 1:

$$a_j(x) = b_j + \sum_{i=1}^h w_{ji} z_i(x), \quad z_i(x) = \tanh \left( c_i + \sum_{k=1}^d v_{ik} x_k \right) \quad (6)$$

The number of hidden units  $h$  controls the capacity (the number of parameters and the flexibility of the model) and  $d$  is the dimension of the input  $x$ . By convention, setting  $h = 0$  results in a linear model ( $a_j(x) = b_j + \sum_{k=1}^d w_{jk} x_k$ ).

The transfer function at the output of the neural network is chosen so as to impose range constraints on the parameters of the mixture. The mixture weight  $\pi_i(x) = P(i|X = x)$  is the probability that the  $i^{\text{th}}$  component is responsible for generating  $y$  given  $x$ . It must therefore be positive and all the  $\pi_i(\cdot)$ 's must sum to one. This is ensured by a *softmax* function:  $\pi_i(x) = \exp(a_i(x)) / \sum_j \exp(a_j(x))$ , where the  $a_j(\cdot)$ 's,  $j = 1 \dots m$  are the neural network outputs dedicated to the priors  $\pi_j(\cdot)$ 's. A *softplus* function ( $\text{softplus}(x) = \log(1 + e^x)$ ) is used to guarantee the positivity of the  $\sigma_i(\cdot)$ 's. The *softplus* has been introduced by [8]; like the exponential, the *softplus* has a positive range but it grows slower than the exponential which makes numerical optimization more stable<sup>2</sup>. In the experiments we have also constrained the  $\xi_i(\cdot)$ 's to be positive with a *softplus*. The  $\mu_j(\cdot)$ 's are unconstrained  $a_j(\cdot)$ 's.

<sup>2</sup>Note that if  $x > 0$ , we have  $\text{softplus}(x) = x + \log(1 + e^{-x})$  and asymptotically we have that  $\lim_{x \rightarrow \pm\infty} \text{softplus}(x) \rightarrow x^+$  where  $x^+$  denotes the positive part of  $x$

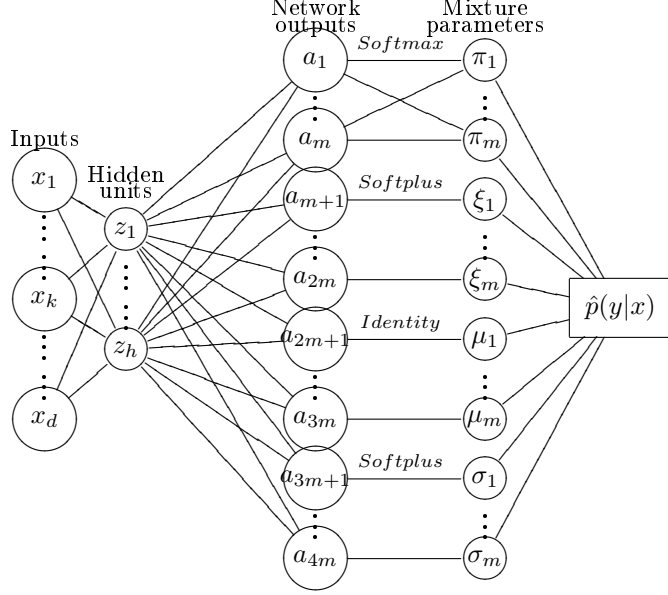


Figure 1: *Conditional mixture model: a feed-forward neural network with one hidden layer and hyperbolic tangent activation function is used to predict input dependent mixture parameters. Appropriate transfer functions at the network outputs are used to impose range constraints.*

The free parameters of the conditional mixture model are thus the neural network parameters  $\Theta = (b, c, v, w)$ . These are determined by minimizing the empirical negative log-likelihood:

$$l(\Theta) = - \sum_i^n \log \hat{p}(y_i|x_i). \quad (7)$$

For each example, we obtain the gradient of the empirical negative log-likelihood with respect to  $\Theta$  through  $\frac{\partial l}{\partial \Theta} = \sum_i \frac{\partial l}{\partial a_i} \frac{\partial a_i}{\partial \Theta}$ , with  $\frac{\partial l}{\partial a_i}$  computed as follows, with  $\varphi$  be the mixture parameters for input  $x$ :

- If  $1 \leq j \leq m$ ,  $a_j$  is one of the outputs controlling the priors and its derivative can be expressed as:

$$\frac{\partial l}{\partial a_j} = \frac{\partial l}{\partial \hat{p}(y|\varphi)} \sum_{k=1}^m \frac{\partial \hat{p}(y|\varphi)}{\partial \pi_k} \frac{\partial \pi_k}{\partial a_j}. \quad (8)$$

- On the other hand, if  $m+1 \leq j \leq 4m$ ,  $a_j$  governs one of the hybrid Pareto component parameter  $\varphi_j$  and its derivative is simpler:

$$\frac{\partial l}{\partial a_j} = \frac{\partial l}{\partial \hat{p}(y|\varphi)} \frac{\partial \hat{p}(y|\varphi)}{\partial \varphi_j} \frac{\partial \varphi_j}{\partial a_j}. \quad (9)$$

In both equations we have  $\frac{\partial l}{\partial \hat{p}(y|\varphi)} = -\frac{1}{\hat{p}(y|\varphi)}$ . When the derivative is taken with respect to one of the

mixture weights, we have, for  $1 \leq j \leq m$ :

$$\frac{\partial \hat{p}(y|\varphi)}{\partial \pi_j} = h(y; \xi_j(x), \mu_j(x), \sigma_j(x)).$$

Differentiating with respect to the hybrid Pareto parameters  $\varphi_j$ , for  $m + 1 \leq j \leq 4m$  and  $i = j \bmod 3$ ,

$$\frac{\partial \hat{p}(y|\varphi)}{\partial \varphi_j} = \pi_i(x) \frac{\partial}{\partial \varphi_j} h(y; \xi_i(x), \mu_i(x), \sigma_i(x)).$$

The derivative of the priors and of the mixture parameters with respect to the network outputs are

$$\begin{aligned} \frac{\partial \pi_k}{\partial a_j} &= \begin{cases} \pi_j(1 - \pi_j) & \text{if } j = k \\ -\pi_k \pi_j & \text{if } j \neq k \end{cases} & \frac{\partial \varphi_j}{\partial a_j} &= 1 & \text{if } \varphi_j = \mu_i(x) \\ \frac{\partial \varphi_j}{\partial a_j} &= 1 - \exp(-\xi_i(x)) & \text{if } \varphi_j = \xi_i(x) & \frac{\partial \varphi_j}{\partial a_j} &= 1 - \exp(-\sigma_i(x)) & \text{if } \varphi_j = \sigma_i(x). \end{aligned}$$

## 6 Experiments

We compared our conditional mixture of hybrid Paretos (**CMM-H**) with two types of benchmarks: conditional mixture models (also with a neural network to predict parameters) with different types of components (**CMM-G** for Gaussian, **CMM-T** for Student t and **CMM-L** for Log-Normal), and non-parametric multivariate models on  $(X, Y)$  (Gaussian mixture **MM-G** and Parzen window estimator **MM-P**), transformed to estimate  $p(Y|X) = p(X, Y)/p(X)$ .

The conditional mixture models are all trained by conjugate gradient descent to minimize the negative log-likelihood. The multivariate mixture of Gaussians is trained by the EM algorithm. Since the optimization may lead to local minima, during learning, each mixture is re-initialized randomly 5 times and the optimization is re-started accordingly (this is done only on the real data sets). We keep the parameters that gave the smallest training error.

The conditional mixture models (regardless of the type of components), have two hyper-parameters: the number of hidden units  $n_h$  for the neural network and the number of components  $m$  of the mixture. The multivariate mixture of Gaussians has one hyper-parameter, the number of components. The variance-covariance matrix is chosen to be diagonal except for the experiments on the artificial data sets where the matrix is full. The variance-covariance matrix for the multivariate Parzen window estimator has two hyper-parameters,  $\lambda_x$  which controls the input variance and  $\lambda_y$  which controls the target variance.

### 6.1 Artificial data sets

We generated data from a conditional Fréchet distribution whose parameters are made conditionally dependent on the input by using either a linear or a sine-shaped functional. The tail index of the Fréchet was chosen to be either in the interval  $[1/6, 1/4]$  to allow for moderately heavy tails or in the interval  $[1/2, 2]$  to allow for heavier tails. We have thus four distinct generative models.

For this experiment, the training and the test set both had 500 observations. To capture the performance relative to the generative model, we measure the performance with the out-of-sample relative log-likelihood:

$$\mathcal{RLL}(\mathcal{D}) = - \sum_{i=1}^n \log \left( \frac{p(y_i|x_i)}{\hat{p}(y_i|x_i)} \right),$$

where  $p(\cdot)$  is the density function of the generative model,  $\hat{p}(\cdot)$  is the density function of the estimator and the sum is over the test set  $\mathcal{D}$ . The smaller the RLL criterion is, the better the estimator is performing. We generated 20 pairs of training and test sets. Table 1 presents a sample overview of the results; it shows the average out-of-sample RLL along with its standard error over the 20 test sets. The generative model is the conditional Fréchet with linearly dependent parameters and moderately heavy tail ( $\xi \in [1/6, 1/4]$ ). All conditional mixture models have one hidden unit (which theoretically should be sufficient since the functional dependence is linear) and the number of components is allowed to increase.

Table 1: Average out-of-sample RLL (standard err.) between predicted density of the estimators and the generative model - conditional Fréchet distribution. *Smaller values mean better estimators.*

$m$	CMM-H	CMM-G	$m$	MM-G
1	10.0 (6.6)	161.6 (28.3)	1	179.1 (28.9)
2	11.0 (6.0)	53.9 (20.7)	2	202.8 (96.8)
4	9.8 (5.4)	36.8 (21.7)	4	187.9 (68.1)
8	11.9 (5.3)	29.6 (13.5)	8	221.0 (86.0)
$m$	CMM-T	CMM-L	$(\lambda_x, \lambda_y)$	MM Parzen
1	134.9 (36.8)	112.7 (18.1)	$(10^{-3}, 10^{-3})$	292.3 (53.8)
2	37.3 (14.5)	37.6 (25.9)	$(10^{-3}, 10^{-2})$	130.8 (37.1)
4	30.4 (16.8)	19.3 (9.0)	$(10^{-2}, 10^{-3})$	307.0 (52.4)
8	30.8 (15.8)	20.1 (8.3)	$(10^{-2}, 10^{-2})$	228.2 (34.7)

Table 1 shows that the conditional mixture with hybrid Pareto components has the smallest RLL even with only one component in the mixture. The complete results for all four data sets give a similar insight; the complete set of results for this data set is provided in the Appendix.

## 6.2 Insurance data set

In a second set of experiments we used real insurance data graciously provided by an anonymous insurance company. The complete distribution of the claims include a mass point at 0. One way to deal with this is to use a probabilistic classifier that predicts, given a client profile  $X$ , the most probable class (claim = 0, claim > 0). For the second class, we need to estimate  $p(\text{claim}|X, \text{claim} > 0)$  and this is the part of the problem we addressed here. This is why the records used in the experiments are only for policies that had a non-zero claim. Data from one year, containing 54119 records with positive claims, were used for training, hyper-parameter selection, testing and model comparison. The dependent variable  $Y$  is the claim amount divided by the duration of the policy. The input variable  $X$  is a vector of 140 numbers, mostly binary indicators, describing the client profile. The numeric inputs have been standardized. Principal component analysis has been applied on the input variable to reduce dimensionality: enough components (between 61 and 69 depending on the training set size) were retained to explain 90% of input variance. The

histogram of the positive claims smaller than 5000\$ of Figure 2 illustrates the unconditional distribution; it shows that the distribution has at least two modes. This distribution is strongly skewed: more than 75% of the claim amounts are smaller than the average claim amount.

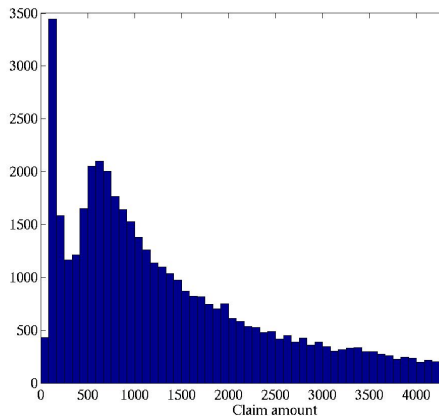


Figure 2: *Insurance data set: histogram of the positive claims smaller than 5000\$.*

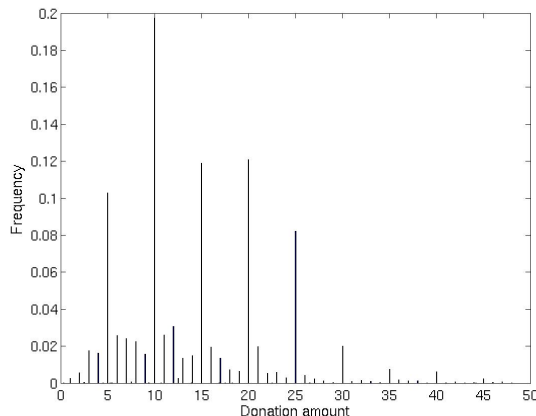


Figure 3: *KDD cup 98 data set: bar plot of donation amounts smaller than 50\$.*

The performance is measured by comparing how a competing estimator performs relative to the proposed conditional mixture of hybrid Paretos. Let  $(x, y)$  be a particular data point, the relative performance measure is then written as:  $\mathcal{R}(x, y) = \log(\tilde{p}(y|x)) - \log(\hat{p}(y|x))$  where  $\tilde{p}(\cdot)$  is the density function of the conditional mixture of hybrid Paretos and  $\hat{p}(\cdot)$  is the density function of a competing estimator. Positive values indicate that the conditional mixture of hybrid Paretos performed better than the competing estimator.

Increasing sizes of training sets were used; the validation set was fixed as the quarter of the size of the training set and was used for hyper-parameters selection. The remaining data was used for testing and model comparison. The average relative performance on the test set with its standard error in parentheses are given in table 2 for all training set sizes. The hyper-parameters selected on the validation set are given in table 3.

Table 2: Average **relative performance** (standard err.) in test with respect to CMM-H for the insurance data set,  $n$  being the training set size. Positive values indicate that the CMM-H performed better.

$n$	CMM-G	CMM-T	CMM-L	MM-G	MM-P
400	93 (43)	0.73 (0.0057)	0.037 (0.018)	0.67 (0.081)	67 (58)
800	246 (90)	0.58 (0.005)	0.021 (0.0043)	0.69 (0.069)	67 (58)
1600	46 (19)	0.68 (0.0043)	0.0014 (0.0068)	0.85 (0.082)	69 (59)
3200	21 (9.8)	0.59 (0.0039)	0.059 (0.011)	0.92 (0.082)	698 (594)
6400	72 (30)	0.44 (0.0034)	0.027 (0.013)	0.96 (0.063)	5.3 (4.5)

Table 3: *Hyper-parameters selected on the validation set for the insurance data set.*

$n$	CMM-H	CMM-G	CMM-T	CMM-L	MM-G	MM-P
400	(1, 14)	(1, 10)	(1, 14)	(1, 14)	4	(100, 1000000)
800	(1, 10)	(1, 12)	(1, 12)	(1, 18)	4	(100, 1000000)
1600	(1, 16)	(1, 10)	(1, 14)	(1, 14)	4	(100, 1000000)
3200	(1, 12)	(1, 8)	(1, 18)	(1, 16)	6	(100, 100000)
6400	(1, 18)	(1, 10)	(1, 8)	(1, 14)	4	(100, 10000000)

We see in table 2 that the performance of the conditional mixture with Gaussian components and of the multivariate Parzen window estimator is really poor in two instances. This is because these two algorithms are greatly affected by the presence of previously unseen extremes in the test set. The conditional mixture with hybrid Pareto components is steadily outperforming the competing algorithms although, in some cases, its performance is not significantly better than the conditional mixture with Log-normal components. This could be explained by the fact that the Log-normal tail is a particular case of generalized Pareto tail.

### 6.3 KDD cup 98 data set

In the last set of experiments, we used the data set provided by the Fourth International Conference on Knowledge Discovery and Data Mining (KDD Cup 98). The dependent variable  $Y$  is the amount donated to a national veterans organization. The input variable  $X$  has 479 fields describing the donor profile. A binary variable indicates whether or not a person responded to the promotion; the donation amount is only observed when this variable is on. Just like for the insurance data set, a probabilistic classifier could be used to predict, given  $X$ , the probability that the person will make a positive donation. However, we addressed only the problem of estimating  $p(Y|X, Y > 0)$ . We thus have a total of 9716 positive records. We note that 75% of the donations are less or equal to 20\$ although some donations go all the way up to 500\$. The target variable takes value in a discrete set containing mainly integer numbers; the amounts corresponding to multiple of 5\$ are especially frequent as can be seen from the bar plot of Figure 3.

We followed [9] for preprocessing, yielding five input variables. We used the same procedure as for the insurance data set regarding the performance criterion, the training, validation and test sets. The average performance on the test set along with its standard error is given in table 4 for each competing algorithm and each training set size. The selected hyper-parameters are given in table 5.

Table 4: *Average **relative** performance (standard err.) in test with respect to CMM-H for the KDD cup 98 data set,  $n$  being the training set size. Positive values indicate that the CMM-H performed better.*

$n$	CMM-G	CMM-T	CMM-L	MM-G	MM-P
400	2.3 (0.2)	0.5 (0.021)	1.2 (0.048)	3.1 (0.2)	1515 (2068)
800	1.5 (0.56)	0.37 (0.055)	1.1 (0.056)	2.4 (0.13)	8.3 (12)
1600	0.88 (0.45)	0.26 (0.029)	0.35 (0.043)	1.9 (0.14)	3.1 (0.25)
3200	0.37 (0.092)	0.31 (0.052)	1 (0.052)	2.3 (0.098)	3.9 (2.5)
6400	0.015 (0.1)	0.24 (0.08)	0.72 (0.099)	2.2 (0.11)	1.2 (0.17)

Table 5: *Hyper-parameters selected on the validation set for the KDD cup 98 data set.*

$n$	CMM-H	CMM-G	CMM-T	CMM-L	MM-G	MM-P
400	(1, 18)	(1, 8)	(1, 4)	(1, 16)	16	(100, 0.01)
800	(1, 18)	(1, 18)	(1, 8)	(2, 16)	18	(100, 1)
1600	(1, 4)	(2, 16)	(1, 8)	(1, 18)	20	(1, 100)
3200	(1, 4)	(1, 18)	(1, 18)	(4, 18)	22	(1, 0.01)
6400	(1, 16)	(2, 16)	(1, 8)	(1, 8)	20	(100, 0.01)

The results in table 4 show that for this data set as well, the conditional mixture with hybrid Pareto components outperforms the other algorithms consistently. However, in this case, the closest competitor is the conditional mixture of Gaussians which gives a performance not significantly distinguishable from the conditional mixture of hybrid Paretos when the training set gets larger. This could also be explained by the fact that the Gaussian tail is well approximated by a generalized Pareto tail of index  $\xi = 0$ .

## 7 Conclusion

Fat tailed data are prominent in several commercial applications of statistical machine learning, such as finance and insurance. Research on extreme events has been mainly concerned with unconditional density estimation whereas in many such applications it is required to consider a conditioning variable, which can be very high dimensional. On the other hand, existing tools for representing conditional densities are not always appropriate in the presence of fat tail variations, multi-modal and asymmetric conditional densities. The main contributions of this paper are thus the following.

1. We have introduced a new fat-tailed density, the **hybrid Pareto**, which combines the generalized Pareto with the Gaussian distribution. This density can be used within a mixture model that allows for multi-modal and arbitrarily shaped conditional densities. Such a mixture circumvents the classical problem with the generalized Pareto methodology of determining the appropriate threshold above which the samples are considered to be in the tail.

The hybrid Pareto tail includes as particular cases, the tail of the Gaussian, the Log-Normal and the Student t. By using the hybrid Pareto, we avoid making a specific assumption regarding the heaviness of the tail of the underlying distribution. Also, the asymmetric shape of the hybrid might be more suited for the shape of the generative distribution we are looking at.

2. Second, we have proposed a conditional density model based on such a mixture, whose parameters are learned functions of the conditioning variable. These functions can be parametric or nonparametric and we have worked with a simple neural network formulation, which is flexible enough for many applications.

3. Third, we have shown through a series of experiments on artificial and real data sets that the proposed conditional density modelling provides significant advantages over competing algorithms based on mixtures and nonparametric density estimation.

## References

- [1] Fama E.F. The behavior of stock market prices. *Journal of Business*, 38:34–105, 1965.
- [2] Mandelbrot B. The variation of certain speculative prices. *Journal of Business*, 36:394–419, 1963.
- [3] P. J. Huber. *Robust Statistics*. John Wiley & Sons Inc., 1982.

- [4] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust Statistics, The Approach based on Influence Functions*. John Wiley & Sons, 1986.
- [5] Embrechts P., Kluppelberg C., and Mikosch T. *Modelling Extremal Events*. Applications of Mathematics, Stochastic Modelling and Applied Probability. Springer, 1997.
- [6] Corless R.M., Gonnet G.H., Hare D.E.G., Jeffrey D.J., and Knuth D.E. On the lambert w function. *Advances in Computational Mathematics*, 5:329–359, 1996.
- [7] Bishop C. *Neural Networks for Pattern Recognition*. Oxford, 1995.
- [8] Dugas C., Bengio Y., Bélisle F., Nadeau C., and Garcia R. A universal approximator of convex functions applied to option pricing. In *Advances in Neural Information Processing Systems*, volume 13, 2001.
- [9] Georges J. and Milley A.H. Kdd'99 competition: Knowledge discovery contest. *SIGKDD Explorations*, 1(2), January 2000.

# Appendix

## GPD and hybrid Pareto distribution

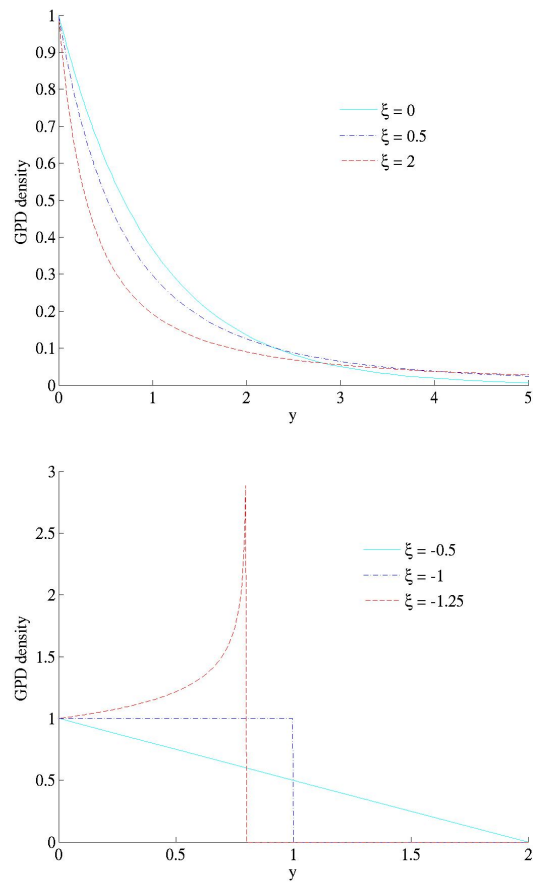


Figure 4: Left panel: GPD density from light ( $\xi = 0$ ) to heavy tail ( $\xi = 2$ ). Right panel: GPD density for finite tails ( $\xi < 0$ ).

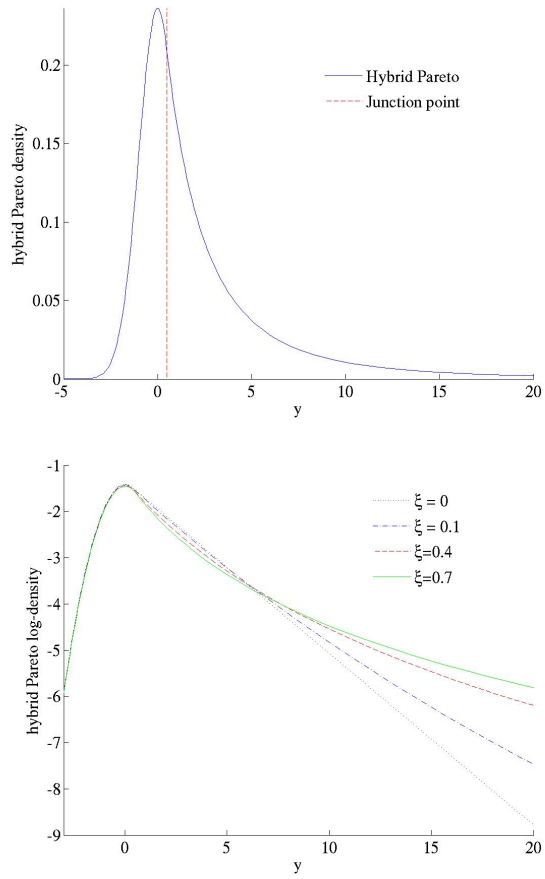


Figure 5: *Left panel: Hybrid Pareto density with parameters  $\xi = 0.4$ ,  $\mu = 0$  and  $\sigma = 1$ . Right panel: Hybrid Pareto log-density for various tail parameters and in all cases  $\mu = 0$  and  $\sigma = 1$ .*

## Maximum-likelihood estimation of hybrid Pareto parameters

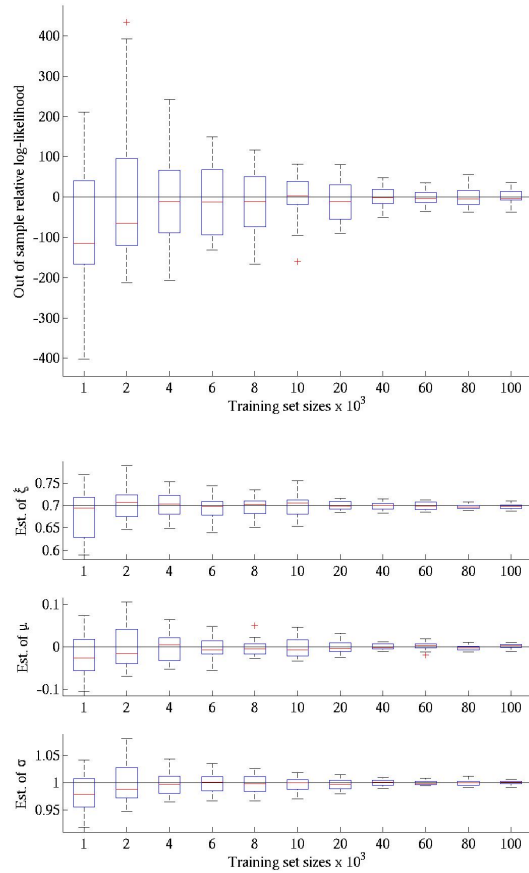


Figure 6: *Estimation of Hybrid Pareto density on data generated by a Hybrid Pareto with parameters  $\xi = 0.7$ ,  $\mu = 0$  and  $\sigma = 1$  as the training set size increases. Left panel: boxplots of out-of-sample relative log-likelihood are converging to zero with respect to the training set size. Right panel: boxplots of maximum likelihood estimators compared to the parameter of the generative density (the line) with respect to the training set size. The estimators are converging to their true parameter values.*

## Pitfalls of the mixture model

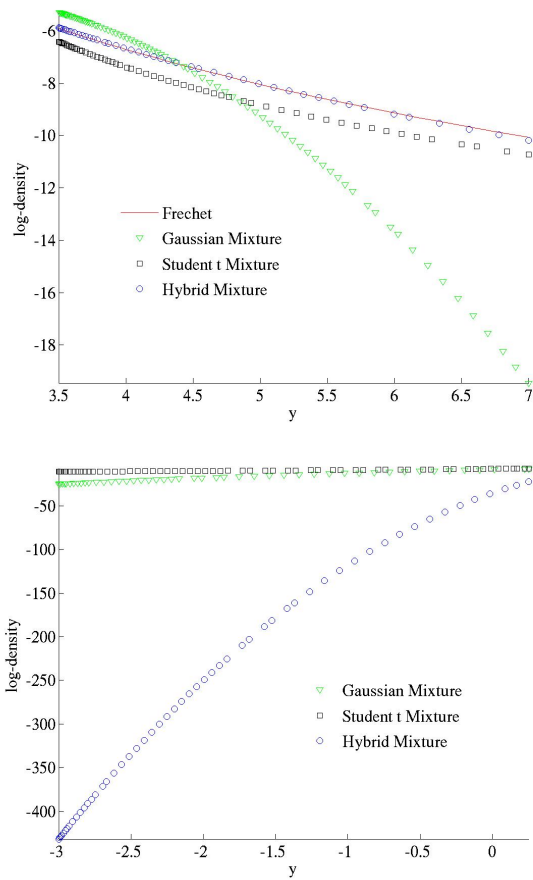


Figure 7: Mixture Estimation with four components comparing Hybrid Pareto, Student  $t$  and Gaussian components; the generative distribution is a Fréchet with tail index  $\xi = 0.2$ , the density being zero for  $y < 0$ . The log-density (vs  $y$ ) of the mixture estimators and of the generative model can be compared for the upper tail in the left panel and for the lower tail in the right panel. We note that the mixture models with Gaussian or Student  $t$  components underestimate the upper tail while they overestimate the lower tail. The mixture model with Hybrid Pareto components is very accurate for the upper tail and its lower tail decreases rapidly.

## Mixture parameter estimation

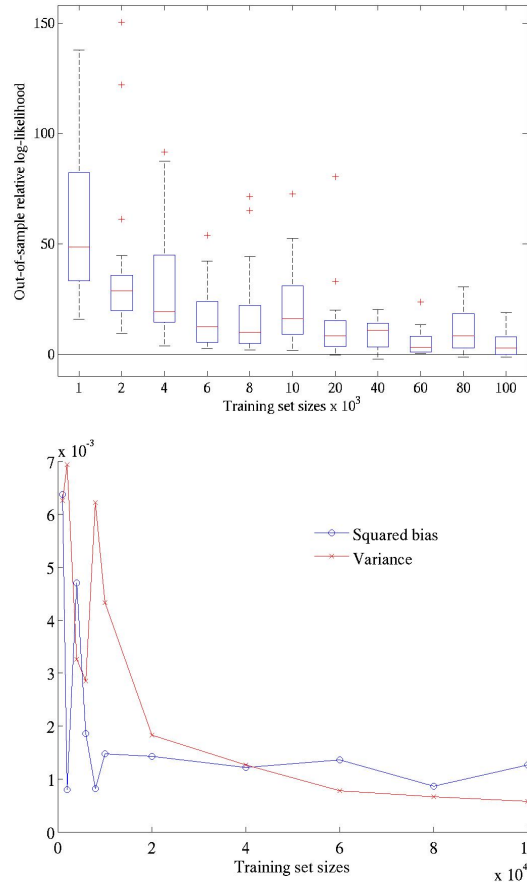


Figure 8: Mixture estimation with Hybrid Pareto components. The generative model is a Fréchet distribution with tail index  $\xi = 0.2$ . The left panel shows boxplots for the out-of-sample relative log-likelihood that decreases with the training set size. The right panel depicts the average squared bias and the variance of  $\xi^* = \max_{1 \leq i \leq m} \xi_i$ , the mixture estimator of  $\xi$ . This suggests that the maximum-likelihood estimation of the mixture of Hybrid Pareto parameters is convergent.

## Experiments on artificially generated data

Here is the complete set of results regarding the experiment on conditional density estimation of fat-tailed data. The generative model is a Fréchet distribution whose parameters are made conditionally dependent on the input by using a linear functional. The tail index of the Fréchet was chosen to be in the interval  $[1/6, 1/4]$  to allow for moderately heavy tails.

Hidden Units	Components	$\mathcal{RLL}_{ave} (\mathcal{RLL}_{std})$	Hidden Units	Components	$\mathcal{RLL}_{ave} (\mathcal{RLL}_{std})$
1	1	10.0403 (6.6355)	1	1	161.6313 (28.3393)
1	2	10.9787 (5.9502)	1	2	53.9171 (20.7171)
1	4	9.8391 (5.4141)	1	4	36.837 (21.7197)
1	8	11.8662 (5.2732)	1	8	29.6483 (13.4835)
2	1	7.9603 (3.9294)	2	1	160.6034 (23.6965)
2	2	9.0772 (5.3939)	2	2	44.5449 (16.8423)
2	4	12.3394 (9.2191)	2	4	32.504 (19.9331)
2	8	14.6452 (7.8258)	2	8	38.8691 (20.2543)
4	1	8.2663 (4.0799)	4	1	161.1916 (27.351)
4	2	9.3567 (5.5425)	4	2	43.7576 (17.3507)
4	4	14.4334 (10.4202)	4	4	50.6303 (37.1841)
4	8	16.8584 (6.6587)	4	8	45.8526 (20.4638)

Table 6: Average out-of-sample relative log-likelihood between the predicted density of the conditional mixture of hybrid Paretos and the generative model along with its standard deviation in parentheses.

Table 7: Average out-of-sample relative log-likelihood between the predicted density of the conditional mixture of Gaussians and the generative model along with its standard deviation in parentheses.

Hidden Units	Components	$\mathcal{RLL}_{ave} (\mathcal{RLL}_{std})$	Hidden Units	Components	$\mathcal{RLL}_{ave} (\mathcal{RLL}_{std})$
1	1	134.9175 (36.7598)	1	1	112.7492 (18.124)
1	2	37.3369 (14.5467)	1	2	37.5882 (25.9494)
1	4	30.3984 (16.7846)	1	4	19.3209 (8.9657)
1	8	30.8234 (15.8305)	1	8	20.1244 (8.2843)
2	1	135.2147 (40.5102)	2	1	111.494 (18.4698)
2	2	43.9366 (19.7032)	2	2	31.4589 (26.0141)
2	4	29.0466 (22.0078)	2	4	17.9546 (8.1212)
2	8	31.9009 (19.231)	2	8	17.6638 (9.2952)
4	1	124.2088 (39.9921)	4	1	111.2322 (17.4709)
4	2	35.196 (17.7019)	4	2	28.3694 (13.8415)
4	4	31.8592 (25.2259)	4	4	16.6944 (9.1476)
4	8	33.9844 (23.2713)	4	8	16.1314 (8.2146)

Table 8: Average out-of-sample relative log-likelihood between the predicted density of the conditional mixture of Student t’s and the generative model along with its standard deviation in parentheses.

Table 9: Average out-of-sample relative log-likelihood between the predicted density of the conditional mixture of Log-Normals and the generative model along with its standard deviation in parentheses.

Components	$\mathcal{RLL}_{ave} (\mathcal{RLL}_{std})$
1	179.0529 (28.8662)
2	202.8453 (96.817)
4	187.8926 (68.137)
8	220.9885 (85.9906)

Table 10: Average out-of-sample relative log-likelihood between the predicted density of the multivariate mixture of Gaussians and the generative model along with its standard deviation in parentheses.

Input Variance	Target Variance	$\mathcal{RLL}_{ave} (\mathcal{RLL}_{std})$
0.0001	0.0001	2210.8914 (190.6821 )
0.0001	0.001	787.2667 (93.1199 )
0.0001	0.01	259.6681 (55.5779 )
0.001	0.0001	800.3513 (132.1046 )
0.001	0.001	292.3278 (53.8444 )
0.001	0.01	130.839 (37.1213 )
0.01	0.0001	585.3545 (118.0382 )
0.01	0.001	307.027 (52.3881 )
0.01	0.01	228.1904 (34.6765 )

Table 11: Average out-of-sample relative log-likelihood between the predicted density of the multivariate Parzen window estimator and the generative model along with its standard deviation in parentheses.